

Grado Universitario en Ingeniería Electrónica Industrial y Automática
2018-2019

TRABAJO FIN DE GRADO

“Chaleco inteligente para medida de ritmo cardíaco”

ÁLVARO ROBLEDO BARBA

Tutora:

CARMEN VÁZQUEZ GARCÍA

Leganés

Julio 2019



Copyright ©2019. Álvaro Robledo Barba

Este documento está publicado bajo la licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional. Para ver una copia de esta licencia, visite <https://creativecommons.org/licenses/by-nc-nd/3.0/es/> o envíe una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, EE.UU. Todas las opiniones aquí expresadas son del autor, y no reflejan necesariamente las opiniones de la Universidad Carlos III de Madrid.



El código vinculado a este documento, publicado en los siguientes repositorios de GitHub <https://github.com/alvroble/AndroidSpeckleBLE> y https://github.com/alvroble/Speckle_RaspberryPi está licenciado bajo la licencia GPLv3. Para ver una copia de esta licencia, visite <https://www.gnu.org/licenses/gpl-3.0.html> o envíe un correo electrónico a gnu@gnu.org.



A mi madre y a mi abuela, que
en paz descansan

Resumen

En los últimos años se ha presenciado un aumento en la importancia de las enfermedades no contagiosas en los países desarrollados, entre las cuales las enfermedades cardiovasculares tienen cada vez más presencia. Ello potencia la necesidad de disponer de diferentes alternativas para monitorizar a los pacientes, o para que las propias personas dispongan de herramientas que les permitan controlar su estilo de vida. Algunos trabajos realizados anteriormente demuestran que el uso de sistemas fotónicos supone una alternativa no invasiva e interesante para la medida de ritmo cardíaco, para su utilización en entornos críticos, entendidos como aquellos en los que se deben evitar las interferencias electromagnéticas, como en la resonancia magnética o en los aviones.

De la misma manera, existe la necesidad de disponer de alternativas que estén alineadas con la siguiente generación de comunicaciones, 5G, en el marco del internet de las cosas o *Internet of Things* (IoT), con el fin de que el usuario pueda interactuar con el sistema de manera sencilla y cómoda.

En el presente trabajo se propone una alternativa no invasiva para la monitorización del ritmo cardíaco, haciendo uso del fenómeno de moteado o *speckle* que tiene lugar cuando se utilizan fuentes de luz coherentes. Para ello, se empleará fibra óptica de plástico (POF, *Plastic Optical Fiber*) como elemento sensor que se puede integrar en un chaleco, se inyectará luz de un láser a la fibra POF y se recogerá la información mediante una cámara en el otro extremo de la fibra. El sistema incluirá un enlace de comunicaciones mediante *Bluetooth Low Energy* (BLE) con una aplicación en Android, donde se podrá observar el ritmo cardíaco del usuario en tiempo real. El presente sistema incluye también cierto nivel de gestión de la información de consumo, pudiendo el usuario conocer el nivel de batería del equipo en todo momento mediante la misma aplicación. Por tanto, el sistema propuesto es un dispositivo integrable en un ponible o *wearable*.

El resultado de este trabajo es un sistema capaz de obtener medidas fiables de vibraciones mecánicas generadas por un zumbador que emula el ritmo cardíaco, con una interacción sencilla por parte del usuario y una comunicación con cierto grado de robustez.

Este proyecto ha sido financiado por varios proyectos: Sensores e Instrumentación en Tecnologías Fotónicas (SINFOTON-CM, Ref: S2013/MIT-2790), SINFOTON2-CM Ref: S2018/NMT-4326 y GREENFIBER Ref: TEC2015-63826-C3-2-R.

Palabras clave: ritmo cardíaco, sensor, fibra óptica de plástico, láser, moteado, ponible, Bluetooth.

Abstract

Last years, non-contagious diseases have increased their presence in developed countries, where cardiovascular diseases have a special relevance. Hence, it is necessary to be able to choose between different alternatives for patient monitoring, or for the people to be able to control their lifestyle. Previous research shows that the use of photonic technology for heart rate monitoring entail an unobtrusive and interesting method for its usage in high electromagnetic interference (EMI) environments, such as magnetic resonance imaging (MRI).

In the same way, it is necessary to propose alternatives consistent with next mobile generation 5G and Internet of Things (IoT), so as there are designed user-friendly systems and applications with good communications capability.

In this project, an unobtrusive heart rate monitoring technique is proposed, based on speckle phenomena from coherent light sources. For this to take place, it is used a plastic optical fiber (POF) as a sensor device that can be embedded in a vest, along with a laser as a light source and a camera as a receiver. This system implements a Bluetooth Low Energy (BLE) communication system which sends the sensory information to an Android application, where the user's heart rate is monitored in real time. Furthermore, an energy management system is designed and implemented, so the user knows at every time the battery level of the system. Overall, the system designed is intended to operate as a wearable device.

The result of this work is a reliable system that can be embedded in a wearable for heart rate-like perturbations monitoring (emulated with a buzz), implementing an easy-to-use user interface and a robust communication.

This project has been partially financed by grants Photonics Technologies for Sensing and Instrumentation (SINFOTON-CM, Ref: S2013/MIT-2790), SINFOTON2-CM Ref: S2018/NMT-4326 and GREENFIBER Ref: TEC2015-63826-C3-2-R.

Keywords: heart rate, sensor, plastic optical fiber, laser, speckle, wearable, Bluetooth.

Agradecimientos

“Si he logrado ver más lejos, es porque me he subido a los hombros de gigantes”, escribía Isaac Newton a su colega Robert Hooke en 1676. En este caso, tengo la suerte de haber pasado una importante etapa de mi vida rodeado de gente maravillosa.

En primer lugar, me gustaría dar las gracias a mis compañeros de la universidad, con quienes he vivido momentos complicados, haciendo trabajos que parecían imposibles y sobre todo aprendiendo juntos esta profesión tan agradecida. Me llevo compañeros y amigos para el resto de mi vida: Meira, Sergio, Rodrigos, Jiménez, Martín, Dani, Pajuelo, Omar, Pablo... gracias a todos.

Este trabajo es fruto de la colaboración con un excelente ingeniero. Gracias Nacho, por romperte la cabeza conmigo en el laboratorio cuando no salían las cosas, y por tus ideas esporádicas que nos han sacado de más de un problema. Te esperan grandes éxitos en tu futuro.

Quiero agradecer enormemente a Dayron por saber transmitirme parte de su extenso conocimiento, y por tener la paciencia de hacerme entender tantas cosas que ni me imaginaba. He aprendido mucho haciendo este trabajo, y gran parte de ello es gracias a ti.

Como no pudiera ser de otra manera, quisiera dar las gracias a mi tutora Carmen, por haber sabido escucharme desde el principio, por haber estado siempre dispuesta a ayudar en todo lo posible, y por darme la posibilidad de trabajar dentro de un gran grupo. Ha sido un verdadero placer trabajar bajo tu tutela.

A los demás estudiantes e investigadores del grupo, y a mis compañeros de laboratorio, quienes siempre han estado dispuestos a echar una mano: Aitor, Sandra, Edu, Julio, Iker, Arantxa, David, Pedro F., Pedro C... gracias.

Por último, me gustaría agradecer a las personas con las que tengo la suerte de compartir mi día a día, empezando por mis amigos y amigas, con quien he compartido toda una vida: Hernán, Guzmán, Marcos, Andrés, Alberto, Edu, Sofía, Esther, María...

A mi pareja Sophie, por saber escucharme en los momentos difíciles y ayudarme a desconectar cuando lo he necesitado. A lo largo de estos años hemos aprendido mucho juntos, gracias por tu cariño y comprensión.

A mi abuelo Vicente, por haberme transmitido su espíritu de superación y constancia, y sobre todo por el humor y la positividad que nos ha contagiado siempre.

A mi hermana María, por su integridad y su responsabilidad. Gracias por tu desinteresada ayuda en casa mientras pasaba días enteros en el laboratorio.

A mi padre Valentín, porque es el mejor ejemplo de vida que habría podido tener. Gracias por dedicar tanto tiempo y esfuerzo en sacarnos hacia delante.

Índice

1. Introducción	1
1.1. Motivaciones.....	2
1.2. Objetivos.....	3
1.3. Marco regulador.....	3
1.3.1. Normativa aplicada a dispositivos médicos.....	4
1.3.2. Normativa aplicada a la electrónica de consumo	5
1.3.3. Normativa aplicada al manejo de láseres	5
1.3.4. Normativa aplicada a la propiedad intelectual	6
1.4. Estructura de la memoria.....	8
2. Estado del arte	9
2.1. Dispositivos <i>wearables</i>	9
2.1.1. Introducción e historia.....	9
2.1.2. Características y componentes principales	11
2.1.3. <i>Wearables</i> en la atención médica	17
2.2. Técnicas de monitorización de ritmo cardíaco.....	18
2.3. Fenómeno de moteado (<i>speckle</i>).....	20
2.3.1. Introducción. Aspectos teóricos.....	20
2.3.2. Aplicaciones y estado de la técnica.....	21
3. Sistema completo.....	25
3.1. Sistema previo.....	25
3.2. Diseño del sistema portátil	28
4. Diseño del hardware	31
4.1. Diseño del procesador	31
4.2. Optoelectrónica	35
4.2.1. Fibra óptica.....	35
4.2.2. Láser	36
4.2.3. Cámara.....	39
4.3. Batería y gestión energética	42
4.3.1. Consumo del sistema.....	42

4.3.2. Diseño de batería	43
4.3.3. Gestión energética	45
4.4. Lógica de funcionamiento.....	48
4.5. Diseño de PCB	50
4.6. Diseño 3D de carcasa para integración	54
5. Diseño del software	57
5.1. Software de procesado de datos	57
5.1.1. Programa de adquisición y pre-procesado.....	58
5.1.2. Programa de procesado.....	61
5.1.3. Comparativa de alternativas de procesado	64
5.2. Software de comunicación	66
5.2.1. Comunicación entre procesos.....	67
5.2.2. Bluetooth Low Energy: Servidor.....	69
5.2.3. Bluetooth Low Energy: Cliente.....	71
6. Medidas y pruebas de integración	75
7. Conclusiones y líneas futuras	83
7.1. Conclusiones	83
7.2. Líneas futuras	84
8. Entorno socioeconómico	85
8.1. Presupuesto.....	85
8.2. Impacto socioeconómico.....	88
8.2.1. Impacto social.....	88
8.2.2. Impacto económico.....	89
9. Planificación global.....	91
Bibliografía.....	93
Anexo I. Esquemático PCB.....	101
Anexo II. Layout PCB	103
Anexo III. Comparativa procesadores	105

Índice de figuras

Figura 1. Arquitectura del sistema previo.	1
Figura 2. Arquitectura del sistema desarrollado.	2
Figura 3. Licencias Creative Commons [10].	7
Figura 4. Tecnología wearable en la monitorización del estilo de vida [12].	10
Figura 5. Aplicación de la tecnología wearable en la atención médica [12].	10
Figura 6. Relación de densidades energéticas de masa y de volumen [16].	13
Figura 7. Diferencias entre MCU y MPU [23].	14
Figura 8. Topología de una red en BLE [24].	16
Figura 9. Esquema de servicios del protocolo GATT [24].	16
Figura 10. Crecimiento previsto de la población a lo largo del siglo [26].	17
Figura 11. Prenda de ropa para asistir la rehabilitación tras un derrame cerebral [28]. .	18
Figura 12. Esquema de funcionamiento de la técnica PPG [31].	19
Figura 13. Señal ECG vs BCG [33].	19
Figura 14. Sistema basado en FBG para monitorización de ritmo cardíaco [34].	20
Figura 15. Patrón de speckle obtenido experimentalmente.	21
Figura 16. Comparación de diferentes técnicas de imagen médica [38].	22
Figura 17. Sistema propuesto por P. J. Pinzón et al. en [1].	23
Figura 18. Configuración más compleja del sistema previo.	26
Figura 19. Configuración más sencilla del sistema previo.	26
Figura 20. Señales obtenidas con el sistema previo.	27
Figura 21. Análisis espectral de una grabación con el sistema previo.	27
Figura 22. Arquitectura general del sistema.	28
Figura 23. Emisión espontánea y emisión estimulada [47].	37
Figura 24. LOC-RM/650/FC0.5 de LasIRvis.	38
Figura 25. Láser utilizado en [1] y [43].	38
Figura 26. Láser de Akozon comprado como alternativa.	39
Figura 27. Láser acoplado a la fibra.	39
Figura 28. Raspberry Pi Zero W y su cámara compatible.	42
Figura 29. Consumo del sistema en diferentes situaciones.	43
Figura 30. Geometría de una celda Li-Ion [57].	44
Figura 31. Batería Li-Po comercial con las dimensiones especificadas.	44
Figura 32. Circuito recomendado para el MCP73831.	45
Figura 33. Relación corriente-Rprog.	46
Figura 34. Fusión entre contador de coulomb y el modelo ModelGauge m5.	46
Figura 35. Circuito recomendado para MAX17260.	47
Figura 36. Circuito de switchover.	48
Figura 37. Sistema completo de carga y monitorización de la batería.	48
Figura 38. Esquema general de lógica de funcionamiento.	49
Figura 39. Dimensiones de la Raspberry Pi Zero W.	50
Figura 40. Layout recomendado para MAX17263.	51
Figura 41. Layout recomendado para MCP73831.	51

Figura 42. Layout diseñado para la PCB.	52
Figura 43. PCB diseñada en visor 3D de KiCad.	52
Figura 44. PCB fabricada en comparación con moneda de euro.	52
Figura 45. Tamaño de la PCB en comparación con Raspberry Pi Zero W.	53
Figura 46. PCB montada conectada a Raspberry Pi.	53
Figura 47. Disposición de los elementos en FreeCAD.	54
Figura 48. Vista en planta de la agrupación de los elementos en FreeCAD.	54
Figura 49. Diseño inicial 3D. Vista cámara.	55
Figura 50. Diseño inicial 3D. Vista planta.	55
Figura 51. Diseño inicial 3D. Vista láser.	55
Figura 52. Diseño inicial 3D. Sujeción batería.	56
Figura 53. Diseño inicial 3D. Carcasa completa.	56
Figura 54. Diagrama de flujo del proceso de adquisición.	58
Figura 55. Diagrama de flujo del procesado de señal.	62
Figura 56. Mecanismo FIFO.	63
Figura 57. Comunicación de procesos en Raspberry Pi Zero W.	67
Figura 58. Esquema aplicación cliente BLE.	69
Figura 59. Captura de aplicación nRF Connect de Nordic Semiconductor.	71
Figura 60. Aplicación cliente en Android. Pantalla de introducción.	72
Figura 61. Aplicación cliente en Android. Pantalla de interacción.	73
Figura 62. Aplicación cliente en Android. Pantalla de información.	73
Figura 63. Caja con zumbador electrónico.	75
Figura 64. Set-up de la recepción en el montaje para las pruebas.	76
Figura 65. Medidas usando diferentes procesados con método de intensidad a 3,5 V. .	78
Figura 66. Solución al error en el conexionado del integrado MAX17263.	79
Figura 67. Setup para medidas con sistema completo.	81
Figura 68. Medidas con variación de frecuencia de excitación.	81
Figura 69. CAGR del mercado de la salud digital según segmento 2015-2020 [92].	89
Figura 70. Planificación global de tareas.	92
Figura 71. Capa de cobre delantera.	103
Figura 72. Capa de cobre trasera.	103

Índice de tablas

Tabla 1. Comparativa entre tipos de baterías para su uso en sistemas portátiles.	12
Tabla 2. Lista de productos considerados para el diseño del procesador.	34
Tabla 3. Familias y productos de referencia de los procesadores a comparar.	34
Tabla 4. Resultados de las pruebas de procesado.	65
Tabla 5. Plan de pruebas de integración.	77
Tabla 6. Rangos de corriente y resolución en función del valor de resistencia [64].	79
Tabla 7. Presupuesto del proyecto desglosado por capítulos.	86
Tabla 8. Resumen presupuesto por capítulos.	87

Lista de acrónimos

BCG – Balistocardiograma

BLE – *Bluetooth Low Energy*, Bluetooth de Bajo Consumo

BPM – *Beats Per Minute*, Latidos Por Minuto

BS – *Beam Splitter*, Divisor de Haz

CAGR – *Compound Annual Growth Rate*, Tasa Anual Compuesta de Crecimiento

cECG – Electrocardiograma capacitivo

CSI – *Camera Serial Interface*

DSP – *Digital Signal Processor*, Procesador Digital de Señal

ECG – Electrocardiograma

FFT – *Fast Fourier Transform*, Transformada Rápida de Fourier

FIFO – *First-In First-Out*

FIR – *Finite Impulse Response*, Respuesta ante Impulso Finita

GATT – *General Attribute Profile*

I²C – *Inter-Integrated Circuit*

IIR – *Infinite Impulse Response*, Respuesta ante Impulso Infinita

IoT – *Internet of Things*, Internet de las Cosas

IRT – *Infrared Thermography*, Termografía por Infrarrojos

LSCI – *Laser Speckle Contrast Imaging*, Imagen por Contraste de Moteado de Láser

MCU – *Microcontroller Unit*, Unidad de Microcontrolador

MPU – *Microprocessor Unit*, Unidad de Microprocesador

OCDE – Organización para la Cooperación y el Desarrollo Económicos

OMS – Organización Mundial de la Salud

PBS – *Polarizing Beam Splitter*, Divisor de Haz sensible a la Polarización

PC – *Personal Computer*, Ordenador Personal

PCB – *Printed Circuit Board*, Placa de Circuito Impreso

PIB – Producto Interior Bruto

POF – *Plastic Optical Fiber*, Fibra Óptica de Plástico

PPG – Fotopletiśmografía

PSD – *Power Spectral Density*, Densidad de Potencia Espectral

RAM – *Random Access Memory*, Memoria de Acceso Aleatorio

SBC – *Single Board Computer*, Placa Computadora

SI-POF – *Step-Index Plastic Optical Fiber*, Fibra Óptica de Plástico de Salto de Índice

SoC – *State of Charge*, Estado de Carga

SPI – *Serial Peripheral Interface*

SSH – *Secure Shell*

STFT – *Short-Time Fourier Transform*, Transformada de Fourier de Tiempo Corto

UUID – *Universally Unique Identifier*, Identificador Único Universal

Capítulo 1

Introducción

El presente trabajo estudia la integración de todos los elementos necesarios para el desarrollo de un sistema *wearable* o “ponible” para la medición del ritmo cardíaco empotrado en un chaleco, mediante el uso de una técnica novedosa y no intrusiva basada en fibra óptica y en la monitorización en el tiempo del patrón de *speckle* o moteado propio de las fuentes coherentes de luz. Se pasará por diferentes aspectos en el diseño del sistema, tanto de *hardware* como de *software*, donde será de crucial importancia una correcta integración y sincronización de todos los componentes para su correcto funcionamiento.

Además, este trabajo pretende servir de continuación de los esfuerzos previamente llevados a cabo por P. J. Pinzón et al., donde se emplea la técnica de *speckle* anteriormente mencionada para monitorizar el ritmo cardíaco de pacientes acostados en un colchón que tiene integrado un cable de fibra óptica [1]. Este sistema también se probó en un chaleco o prenda como elemento sensor con el esquema que se observa en la Figura 1. La luz emitida por un láser se acopla a una fibra óptica de plástico (POF, *Plastic Optical Fiber*) a través de algunos elementos ópticos, atraviesa el elemento sensor y, con una lente se acopla su salida a una cámara comercial. Las medidas de ritmo cardíaco se realizan *offline*, es decir, se graban videos con la cámara que posteriormente son procesados mediante MATLAB.

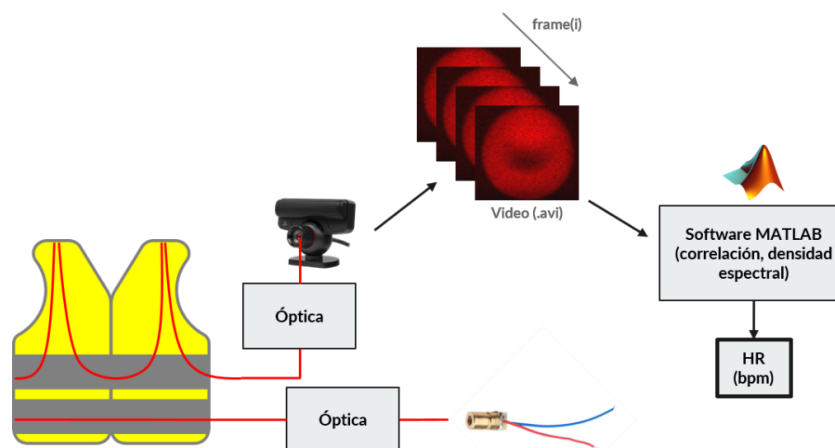


Figura 1. Arquitectura del sistema previo.

En este trabajo se lleva a cabo una integración del sistema de la Figura 1, disminuyendo considerablemente su tamaño, con el fin de que el sistema pueda ser totalmente portátil y opere en tiempo real. Para ello, se desarrolla un sistema basado en microprocesador, y alimentado mediante una batería recargable, que implementa tanto un procesado en tiempo real de la señal del ritmo cardíaco como la comunicación mediante *Bluetooth Low Energy* (BLE) con una aplicación desarrollada sobre Android. Esta arquitectura se resume en la Figura 2, donde el bloque denominado “vibración mecánica” es el dispositivo de ensayo (DUT, *Device Under Test*) que emula las vibraciones inducidas por el ritmo cardíaco, y que en este trabajo se generan con un zumbador electrónico.

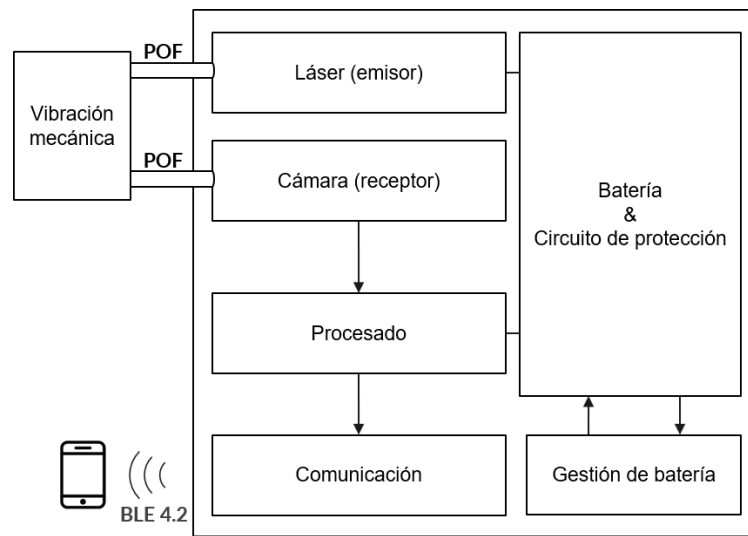


Figura 2. Arquitectura del sistema desarrollado.

1.1. Motivaciones.

Las enfermedades cardiovasculares suponen un problema de creciente importancia en la sociedad actual. Según la Organización Mundial de la Salud (OMS), en el año 2016 se produjeron 17,9 millones de muertes relacionadas con afecciones en el corazón, lo que supone un tercio de todas las muertes producidas en ese año. Estas enfermedades están directamente relacionadas con el estilo de vida actual de las personas: la calidad de la alimentación, la inactividad física, el tabaco o el alcohol son factores de riesgo asociados a la aparición de estas enfermedades [2].

Por otra parte, la atención médica se puede beneficiar de la aparición de tecnologías basadas en el internet de las cosas (IoT), que permiten monitorizar a pacientes de manera remota, de forma que se reduciría la carga de trabajo de los médicos en ciudades como Madrid, donde casi la mitad de los y las profesionales médicos atendieron a más de 40 personas al día en 2015 [3]. Pues la introducción de equipos que monitorizan parámetros de salud de manera remota permite a los médicos llevar un control continuo de la salud de sus pacientes. Permitiendo una reducción del número de consultas, lo que permitiría una mejora general en la calidad de la atención sanitaria.

Sin embargo, se debe asegurar la comodidad del paciente en todo el proceso de la atención médica. Ante la tradicional técnica del electrocardiograma (ECG)—que sin duda es una técnica intrusiva, ya que necesita un contacto directo de los electrodos con la superficie del cuerpo—ha aparecido la fotopletismografía (PPG), una técnica muy empleada en los relojes inteligentes, que emplea luz en el espectro visible para conocer el ritmo cardíaco, iluminando el cuerpo y detectando la absorción o reflexión de la luz. Esta técnica, aunque en menor medida, requiere cierta cercanía a la piel, por lo que resulta interesante desarrollar una nueva técnica donde el contacto sea mínimo, y es en este contexto donde tiene cabida la tecnología basada en fibra óptica, usada de manera extensa para sensar vibraciones estructurales de muy pequeña amplitud. Además, aplicando el análisis del *speckle* o moteado de la luz de un láser que atraviesa la fibra se puede obtener una gran sensibilidad a las vibraciones.

Por lo tanto, el presente trabajo tratará de dar una solución a la introducción del IoT para la monitorización del ritmo cardíaco de los pacientes o de la población en general, mediante una técnica que necesita cierto desarrollo pero que tiene bastantes ventajas.

1.2. Objetivos.

El presente trabajo pretende como objetivo global implementar un sistema portátil que permita la monitorización en tiempo real del ritmo cardíaco utilizando una técnica no invasiva, como es el uso del *speckle*, comprobando diferentes métodos de procesamiento mediante un dispositivo de ensayo.

Para alcanzar el mismo se plantean los siguientes objetivos específicos:

- Diseño y desarrollo del sistema hardware que incluya la capacidad de excitación, sensado, adquisición, y procesado en tiempo real, así como una correcta gestión energética del dispositivo.
- Diseño y desarrollo de una aplicación que permita un control remoto inalámbrico del *wearable*, con un dispositivo Android, basado en un sistema de comunicaciones mediante *Bluetooth Low Energy* (BLE)
- Integración de todos los bloques como un sistema modular portátil y desarrollo de las pruebas que validen la técnica de *speckle* para monitorización de señales cardíacas emuladas con un dispositivo de ensayo.
- Diseño de una carcasa que agrupe los elementos, mostrando la viabilidad del sistema portátil integrando tanto la parte óptica como la parte electrónica del sistema.

1.3. Marco regulador.

En lo referente a la normativa aplicable al sistema que estudia el presente trabajo, ante la falta de normativas referentes a dispositivos *wearables*, se pueden tomar varios puntos de

vista en función de los elementos que componen el sistema y la aplicación final. En primer lugar existe el caso en el que los médicos usarán el sistema aquí desarrollado para monitorizar el ritmo cardíaco de los pacientes, lo que llevaría a interpretar el sistema como un dispositivo médico. Por otra parte, se podría plantear el trabajo en el marco de la electrónica de consumo, más común generalmente para los *wearables*, siendo el usuario final la población en general. Además, el uso de tecnología láser hace necesario el cumplimiento de la normativa que regula el trabajo con fuentes de luz coherentes. Por último, el trabajo desarrollado está publicado bajo determinadas licencias de derechos de autor que se presentarán brevemente.

1.3.1. Normativa aplicada a dispositivos médicos

En cuanto al primer caso, el Parlamento Europeo publicó en abril de 2017 dos reglamentos en referencia a los dispositivos médicos: en primer lugar, el **Reglamento (UE) 2017/745** sobre productos sanitarios (MDR, del inglés *Medical Devices*) [4], y en segunda instancia el Reglamento (UE) 2017/746 para productos sanitarios utilizados para diagnóstico in vitro (IVDR) [5].

En lo que al sistema presentado en este trabajo atañe, se encuentra bajo la regulación de la norma 2017/745, pues el presente dispositivo no se emplea para extraer muestras como sangre o tejidos, tal y como vienen definidos los IVDR en el artículo 2 apartado 2 del Reglamento 2017/746. En este caso, un sistema para monitorización de constantes vitales encaja en el marco del Reglamento 2017/745, que define “producto sanitario” como: “todo instrumento, dispositivo, equipo, programa informático, implante, reactivo, material u otro artículo destinado por el fabricante a ser utilizado en personas, por separado o en combinación, con alguno de los siguientes fines médicos específicos”, siendo el fin acorde con el sistema aquí presentado el de “diagnóstico, prevención, seguimiento, predicción, pronóstico, tratamiento o alivio de una enfermedad”.

Este Reglamento de la Unión Europea responde a la necesidad de actualización de las Directivas 90/385/CEE y 93/42/CEE del Consejo. Organizado en 10 capítulos, el texto propone la regulación de todas las fases del producto sanitario, desde su comercialización y puesta a punto hasta su uso en investigaciones clínicas, pasando por todos los trámites burocráticos que han de llevarse a cabo por las distintas partes que intervienen en el proceso de comercialización (fabricantes, distribuidores, importadores, etc.).

Cada producto ha de cumplir un estricto proceso de evaluación por parte de los organismos reguladores, y satisfacer determinadas condiciones de seguridad, que abarcan todos los aspectos referentes al diseño y a la fabricación, así como posibles peligros a los que pueda estar expuesto el usuario que emplea el producto (radiación, calentamiento...), tal y como se puede leer en el Anexo I [4]. Por otra parte, es importante que cada producto esté debidamente documentado para su utilización, con todas las especificaciones que permitan obtener una mayor transparencia entre fabricante, comerciales y usuarios (Anexo II [4]).

En el Anexo VIII de la norma 2017/745 [4], se especifican las reglas de clasificación de los productos, teniéndose en cuenta la duración del uso (pasajero, a corto plazo o prolongado) y cómo de invasivo es el producto. En este caso, el monitor de ritmo cardíaco propuesto es un producto no invasivo, ya que no existe correspondencia con la definición que se realiza de producto invasivo (“producto invasivo que penetra en el interior del cuerpo a través de la superficie corporal, incluso a través de las membranas mucosas de los orificios corporales por medio o en el contexto de una intervención quirúrgica”).

En el capítulo III del anexo VIII, Regla 4.2, se puede entender que el sistema aquí presentado entraría dentro de la clase I, pues no está destinado a “la conducción o almacenamiento de sangre, líquidos, células o tejidos corporales, líquidos o gases destinados a una perfusión, administración o introducción en el cuerpo”.

Además de cubrir todas las obligaciones y derechos de los distintos agentes económicos involucrados en el proceso, el Reglamento proporciona información acerca de las obligaciones de trazabilidad, evaluación y certificaciones de calidad y conformidad.

Por último, es importante destacar que en el Anexo XV se establecen las normas para llevar a cabo investigaciones clínicas, a partir de los principios éticos reconocidos. En este apartado se establecen las obligaciones del responsable en cuanto a la documentación del proceso de investigación y la finalidad de los datos obtenidos de la misma.

1.3.2. Normativa aplicada a la electrónica de consumo

En el segundo caso, si se estudia la normativa dentro del marco de la electrónica de consumo para tecnologías de la información, las regulaciones se encuentran sobre todo en la protección de datos.

La normativa aplicable en cuanto a protección de datos es el **Reglamento (UE) 2016/679** del Parlamento Europeo, en el marco de la cual el presente sistema “extrae” datos de carácter personal, para lo cual según el artículo 5 de dicho reglamento, se deben tratar los datos de forma lícita, leal y transparente. Para ello, el usuario deberá además dar su consentimiento, de acuerdo a los artículos 6 y 7. Es importante destacar que existe cierta compatibilidad hacia la primera orientación, ya que los dispositivos médicos están necesariamente incluidos dentro del reglamento de protección de datos, de acuerdo con la consideración número 35.

1.3.3. Normativa aplicada al manejo de láseres

Al trabajar con un diodo láser, el sistema aquí desarrollado estará sujeto al **Real Decreto 486/2010**, de 23 de abril, sobre la protección de la salud y la seguridad de los trabajadores contra los riesgos relacionados con la exposición a radiaciones ópticas artificiales [6]. Para radiaciones láser, en el apartado A del Anexo II de dicho Real Decreto se pueden

encontrar los valores límite de exposición. De acuerdo con lo implementado, la radiación emitida se encontraría en el espectro visible, concretamente en una longitud de onda en torno a los 650nm (color rojo). Bajo estas condiciones, se establecen varios límites de exposición:

- Exposición de los ojos al láser ($T_{\text{exposición}} \geq 10\text{s}$): 10 W/m^2 para una apertura de 7mm, para evitar lesiones de la retina.
- Exposición de los ojos al láser ($T_{\text{exposición}} \leq 10\text{s}$): $1,5 \cdot 10^{-4} \text{ W/m}^2$ para una apertura de 7mm y una divergencia de 1mrad.
- Exposición de la piel al láser: $2 \cdot 10^{11} \text{ W/m}^2$ para una apertura de 3,5mm y una duración menor de 1ns, y $2 \cdot 10^3 \text{ W/m}^2$ para exposiciones de larga duración (10^3 - $3 \cdot 10^4 \text{ s}$).

El presente trabajo propone para su desarrollo un diodo láser directamente acoplado a una fibra óptica de plástico, y se asegurará que la luz se mantenga confinada en un lugar cerrado, y que el láser en ningún momento se desprenda de su unión con la fibra. De esta manera se evitarán exposiciones de ningún tipo para el usuario.

1.3.4. Normativa aplicada a la propiedad intelectual

El presente trabajo académico presenta unas implicaciones en cuanto a la distribución de la obra, con el objetivo de proteger los derechos de autor. En España, la propiedad intelectual está regulada por los siguientes escritos:

- Código de Propiedad Intelectual publicado en el Boletín Oficial del Estado (BOE). Actualizado al 15 de abril de 2019 [7].
- Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual [8].

Como alternativa a las licencias *copyright* (“todos los derechos reservados”), aparecen las licencias denominadas *copyleft*, cuya filosofía es la de compartir libremente las obras, siempre que las creaciones derivadas se encuentren bajo la misma licencia que la obra original o equivalente [9]. Por otro lado, desde el año 2004 se cuenta en España con seis licencias del tipo *Creative Commons* [10], mediante las cuales el autor puede escoger cómo compartir su obra, pudiendo decidir todas las condiciones de su distribución.

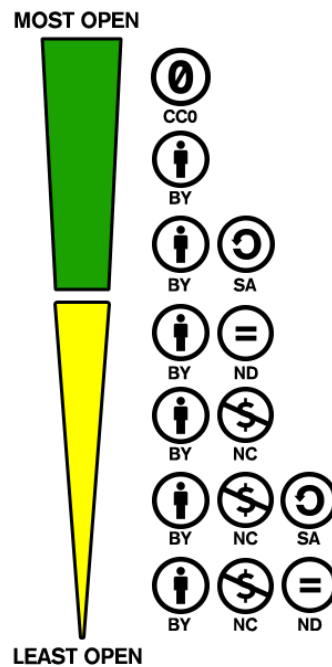


Figura 3. Licencias Creative Commons [10].

En la Figura 3 se pueden observar las distintas combinaciones de licencia ordenadas según la libertad de distribución otorgada por el autor. Las condiciones disponibles son las siguientes [10]:

- **BY (Atribución):** el autor pone la condición de que se le otorgue el crédito en obras que utilicen su trabajo de cualquier manera.
- **SA (Compartir Igual):** se permite copiar, distribuir y modificar la obra siempre y cuando las obras derivadas mantengan las mismas condiciones impuestas por el autor.
- **NC (No Comercial):** las obras derivadas no podrán ser comercializadas salvo aprobación del autor.
- **ND (No Derivadas):** no se permite modificar el trabajo, aunque sí podrá ser copiado y distribuido.

El trabajo que se realiza aquí se publica en acceso abierto en la web de la Universidad Carlos III de Madrid, con el fin de favorecer su libre difusión, bajo una licencia *Creative Commons* Atribución-No Comercial-No Derivadas 4.0 Internacional.

En el caso del software, aparecen otros tipos de licencia de carácter *copyleft*, que permiten proteger los derechos de los desarrolladores de software. La licencia más utilizada es la Licencia Pública General de GNU o GNU-GPL (*General Public License*) [11]. Esta licencia asegura varias libertades para el usuario: libre uso, modificación, libertad para compartir y para compartir los cambios realizados. Por lo tanto, se está hablando de *software* libre. El usuario podrá modificar, copiar, distribuir y beneficiarse económicamente del software desarrollado siempre que todas las obras derivadas sean publicadas también como *software* libre.

El código aquí desarrollado, tanto en Android como en Python, se publicará en la plataforma GitHub bajo la licencia GNU-GPLv3.

1.4. Estructura de la memoria.

En las páginas que siguen, se va a realizar un repaso al proceso completo de diseño del sistema. No obstante, se dará antes una visión general de los conceptos y las técnicas que se relacionan directamente con el trabajo.

En primer lugar, en el capítulo 2 se presenta el estado del arte de los dispositivos *wearables* en general, realizando un análisis de su historia y de sus principales componentes en cuanto a electrónica y comunicaciones, donde se pasará por varios de los puntos que más tarde se tratarán en el diseño. Tras ésto, se realizará un recorrido por el estado de la aplicación de la tecnología *wearable* en la atención médica.

De la misma forma, se comentará también el estado de la técnica en cuanto a la medida del ritmo cardíaco, pasando después a introducir la tecnología de *speckle* o moteado y sus aplicaciones, principalmente orientado a la monitorización de constantes vitales.

En el capítulo 3, se presenta el diseño del sistema a alto nivel. En primer lugar, se desarrollará con mayor detalle el sistema previo, para después comentar los aspectos más relevantes y especificaciones del sistema a nivel de bloques.

Posteriormente, se ha decidido dividir el diseño en dos bloques: hardware y software. Dentro del hardware, en el capítulo 4 se encontrarán diferentes apartados dedicados tanto a la electrónica y su acondicionamiento, como a la parte de óptica y optoelectrónica, y el diseño mecánico de la carcasa. En el bloque de software, en el capítulo 5, también se dividirá en dos: el procesamiento de señal y las comunicaciones. En este caso se hablará tanto del software del sistema desarrollado como de la aplicación implementada para monitorizar el ritmo cardíaco de forma remota.

Posteriormente, en el capítulo 6 se comentará la integración final del sistema, así como las pruebas realizadas sobre el sistema completo. Tras esto, en el capítulo 7 se presentarán las conclusiones obtenidas y las propuestas de líneas de trabajo a seguir en el futuro.

Por último, el trabajo quedará completo con un análisis socioeconómico del mismo en el capítulo 8, además de mostrar la planificación de tareas que se ha seguido a lo largo de su realización en el capítulo 9.

De manera adicional, se añaden los anexos que se han considerado oportunos para complementar el presente documento.

Capítulo 2

Estado del arte

En este capítulo se pretenden introducir los diferentes temas de interés que rodean el presente trabajo. En primer lugar, se hablará sobre los dispositivos wearables, orientando su aplicación al sector salud. Tras esto, se pasarán a comentar algunas de las diferentes técnicas de monitorización de ritmo cardíaco más utilizadas. Por último, se realizará un análisis del principio físico en el que se basa el sistema aquí desarrollado, el patrón de moteado o *speckle*.

2.1. Dispositivos *wearables*.

2.1.1. Introducción e historia.

La palabra inglesa “wearable” (que tiene la capacidad de ser llevado como prenda de ropa) y la palabra “tecnología” (aplicación de conocimientos en ciencia e ingeniería para su aprovechamiento práctico), llevan a asumir que el primer dispositivo wearable en la historia fueron las gafas, invento que data en torno al año 1286. Otros ejemplos de tecnología *wearable* que aplicaban la ciencia para dar soluciones a problemas prácticos hace varios siglos son el Huevo de Nuremberg, el primer reloj portátil, en torno al año 1510; y el anillo de ábaco utilizado para cálculos aritméticos en el siglo XVII en China.

No fue hasta finales del siglo XX, con el imparable crecimiento de la tecnología electrónica, que comenzaron a aparecer dispositivos portátiles con cierto nivel de inteligencia y prestaciones, empezando por relojes-calculadora como el Pulsar Calculator Watch, o dispositivos para reproducir audio como el tradicional Sony Walkman [11]. Hoy en día, la electrónica ha asimilado el término anglosajón “*wearable technology*”, que ha pasado a referirse únicamente a dispositivos electrónicos incorporados en la ropa y accesorios que pueden llevarse en el cuerpo.

A lo largo de los últimos años, la aparición de tecnologías como el IoT o el *Big Data*, el tamaño cada vez más pequeño de los componentes electrónicos y la aparición de nuevas técnicas de sensado, han permitido el desarrollo de dispositivos con el potencial de su uso en múltiples aplicaciones en la vida cotidiana del ser humano. El deporte y la salud son

los campos donde se han producido más desarrollos específicos de la tecnología *wearable*, debido a la importancia de la medida de parámetros biológicos y el análisis de datos empleado para evaluar el estilo de vida de la población, siendo por tanto una herramienta tanto para personas que buscan mejorar su calidad de vida como para los doctores que quieren monitorizar a sus pacientes. En la Figura 4 y la Figura 5 se puede obtener una visión general de la aplicación de esta tecnología en el deporte y la atención médica, respectivamente [12]. No obstante, el desarrollo de dispositivos tan versátiles como los relojes inteligentes (*smartwatches*) o las gafas inteligentes (*smart glasses*), han llevado la tecnología *wearable* a cientos de millones de personas, con aplicaciones en educación, finanzas, entretenimiento o transporte.

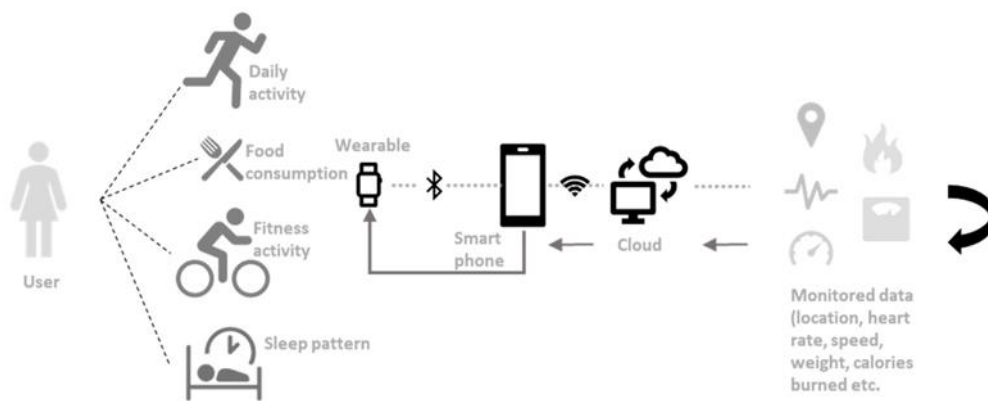


Figura 4. Tecnología wearable en la monitorización del estilo de vida [12].

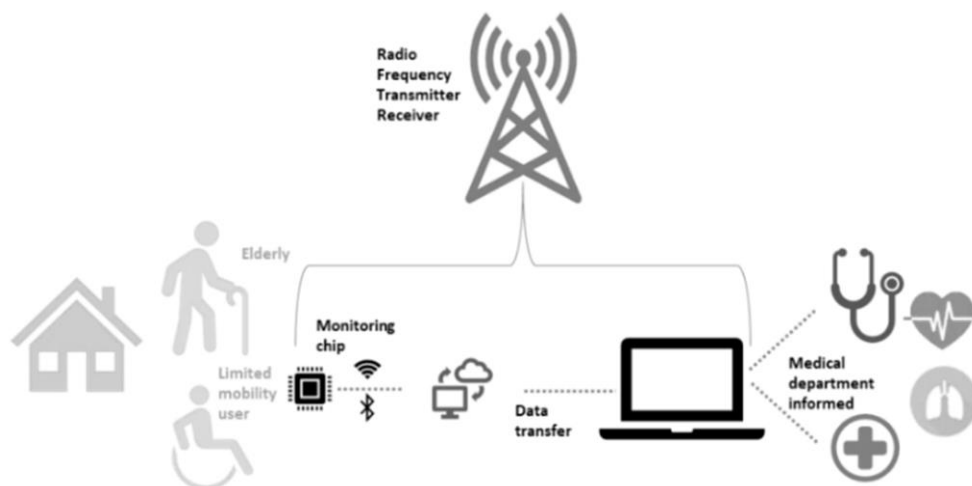


Figura 5. Aplicación de la tecnología wearable en la atención médica [12].

Los dispositivos *wearables* como las pulseras, los relojes o las gafas inteligentes, además de las gafas de realidad virtual, basan sus funciones en avances tecnológicos de vanguardia, y es por ello por lo que se trata de un mercado en constante movimiento. Generalmente, estos dispositivos disponen de un microprocesador que se encarga de recoger toda la información procedente de los diferentes sensores y procesarla, llevando

a cabo un análisis o comunicando directamente esta información a otro dispositivo que tenga la capacidad de analizar los datos. La comunicación se suele realizar mediante tecnologías inalámbricas como Wi-Fi, Bluetooth o LTE 4G. Los desarrollos futuros en el marco de las comunicaciones 5G aseguran un ancho de banda mayor para los dispositivos que intercambian volúmenes elevados de información, además de una latencia de apenas 1ms, introduciendo mejoras importantes para el futuro de la comunicación de los dispositivos *wearables* [13].

2.1.2. Características y componentes principales

Debido al temprano desarrollo de los dispositivos *wearables*, no existen por el momento unas propiedades bien definidas. Si bien, existe una gran variedad de productos y se observa que todos ellos poseen como aspectos fundamentales los que se indican a continuación [14]:

- **Comodidad**

Al tratarse de dispositivos que se llevan como accesorios o prendas de ropa, no deben molestar al usuario, lo que significa una reducción en el peso y el tamaño del sistema.

Además, los sistemas deben cumplir con un cierto nivel alto de abstracción en lo relativo a su utilización, es decir, deben proponer una interacción sencilla con el usuario, de manera que cualquier persona pueda emplear el dispositivo. En este apartado se deben tener en cuenta otras cuestiones como la accesibilidad y la conectividad del sistema, en función de la aplicación.

Esta característica que engloba los conceptos de interacción con el usuario, peso, tamaño y accesibilidad, será el elemento que condicione otra de las propiedades de los dispositivos, como es la gestión energética.

- **Gestión energética**

Es evidente que los dispositivos *wearables* deben estar correctamente alimentados por baterías que, de cara a aumentar la sostenibilidad del producto, deben ser recargables. La capacidad y el tamaño de las baterías disponibles en el mercado están condicionadas por varios parámetros: el consumo del sistema, el tiempo de duración especificado, el voltaje de funcionamiento o la geometría del dispositivo, entre otros.

En el mercado existen diversos tipos de baterías:

- NiCd (Níquel-Cadmio).
- NiMH (Hidruro de Níquel-Metal).
- Plomo Ácido.
- Li-Ion (Ion de Litio).

– Li-Po (Polímero de Litio).

En la Tabla 1 se presenta una comparación entre ellas teniendo en cuenta las características que se consideran más relevantes a la hora de diseñar un sistema portátil [15].

Tabla 1. Comparativa entre tipos de baterías para su uso en sistemas portátiles.

	NiCd	NiMH	Plomo Ácido	Li-Ion	Li-Po
Voltaje de celda (nominal)	1,25V	1,25V	2V	3,6V	3,6V
Corriente de descarga					
Pico	20C	5C	5C	2C	2C
Óptima	1C	≤0.5C	0.2C	≤1C	≤1C
Densidad energética					
Gravimétrica (Wh/kg)	45-80	60-120	30-50	110-160	100-130
Volumétrica (Wh/L)	80-180	100-280	60-75	230-370	230-370
Temperatura de operación (en descarga)	-40 a 60°C	-20 a 60°C	-20 a 60°C	-20 a 60°C	0 a 60°C
Autodescarga / mes	20%	30%	5%	10%	10%
Mantenimiento	30-60 días	60-90 días	3 a 6 meses	NO	NO
Tolerancia a sobrecarga	moderada	baja	alta	muy baja	baja
Toxicidad	alta	baja	alta	baja	baja
Seguridad	estable (fusibles)	estable (fusibles)	estable	necesaria protección	necesaria protección

Por otra parte, observando la Figura 6 se puede entender por qué las baterías de litio son las más empleadas no sólo para *wearables*, sino para dispositivos portátiles en general, ya que permiten almacenar una mayor cantidad de energía para un tamaño y peso dados.

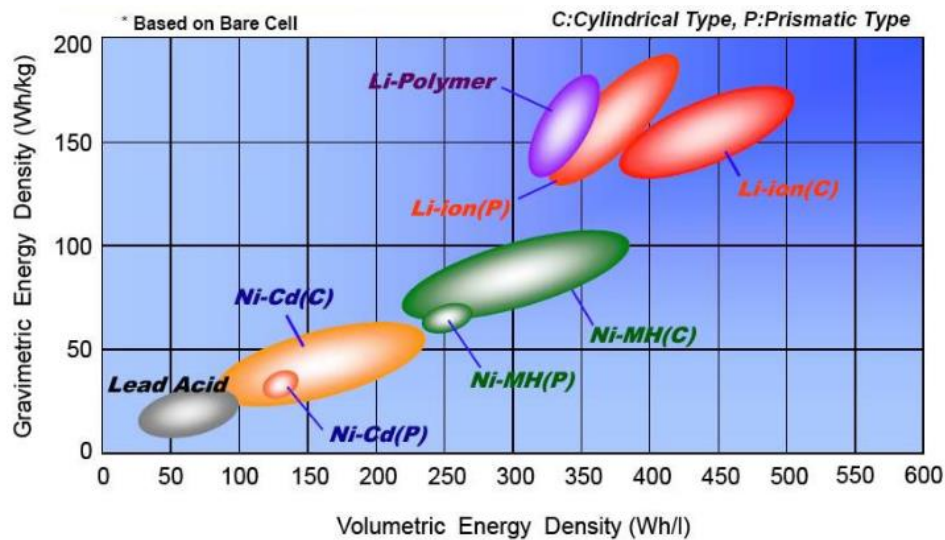


Figura 6. Relación de densidades energéticas de masa y de volumen [16].

Dado el tamaño limitado de las baterías de los dispositivos *wearables*, en los últimos años se han implementado tecnologías que ayudan a gestionar el consumo de recursos y la carga de la batería, entre ellas el llamado *Energy Harvesting* (recolección de energía), a partir de fuentes de energía como el propio calor del cuerpo o la luz ambiente [17].

Además, aparecen diversos sistemas integrados que cada vez disponen de un mayor número de funcionalidades en cuanto a gestión de energía, lo que ayuda al diseño de dispositivos cada vez más pequeños. Estos circuitos integrados pueden ir desde protecciones para las baterías, necesarias en algunos casos para proporcionar la seguridad del usuario aislando la carga, a cargadores de baterías, que son capaces de realizar el proceso de carga completo de las baterías cuando se conecta una fuente de alimentación [18].

Pero sin embargo son los llamados *Fuel Gauges* los dispositivos que más funcionalidades ofrecen. Se tratan de, como bien indica su traducción, “indicadores de combustible”, recordando así al indicador presente en los vehículos tradicionales. Estos dispositivos, en electrónica se han empleado tradicionalmente en el sensado de la corriente que pasa por una resistencia, con una posterior integración, lo que recibe el nombre de “contador de coulomb”. Sin embargo, esta integración introduce errores y, aunque sea un método que funciona relativamente bien, se han introducido mejoras en los dispositivos, que tratan de combinar el contador de coulomb con el cálculo a partir de un modelo matemático correspondiente a la relación voltaje-carga de la batería [19]. La necesidad de emplear electrónica digital para realizar esta computación ha proporcionado a estos circuitos integrados la capacidad de almacenar y monitorizar el estado de carga (SoC, *State of Charge*) de la batería incluso cuando el sistema huésped (el procesador) está apagado, presentando un consumo despreciable para la batería. Este almacenamiento permite además realizar una

supervisión de diferentes parámetros que puedan ser necesarios para monitorizar la vida de la batería. En el apartado de diseño correspondiente se profundizará más en estos dispositivos.

- **Procesador**

El tipo de procesador que suelen emplear estos dispositivos presenta tanta variedad como sus aplicaciones. En primera instancia, se pueden encontrar soluciones basadas en microcontrolador (MCU), como la plataforma FLORA de Adafruit [20], o la serie de ultra bajo consumo de MCUs MAX326XX de Maxim [21]. Estos dispositivos, aunque no tienen implementado ningún sistema operativo, solamente pueden ejecutar un único bucle de control a la vez. Evidentemente, esto trae ventajas como la disminución en el consumo, pero también problemas de limitación de funcionalidades. Esta propiedad hace que dentro del chip del microcontrolador deban estar integrados todos los elementos periféricos.

Por otra parte, existen los sistemas basados en microprocesador (MPU), con una mayor capacidad de procesar la información, que permiten una gran versatilidad a costa de aumentar la frecuencia de reloj y, por tanto, el consumo. Además, los periféricos y los elementos de comunicación se encuentran en el “exterior” del dispositivo. Esto hace que estos sistemas puedan implementar un sistema operativo que realice la gestión de diferentes procesos simultáneos. Los ejemplos son innumerables, destacando fabricantes como Arm y Qualcomm. Además, como se puede consultar en [20], existen placas de desarrollo disponibles que son versátiles de cara al diseño y prototipado de productos. Una placa basada en MPU podría ser la Raspberry Pi en su versión Zero [22].

En la Figura 7 se puede comprobar la diferencia entre MCU y MPU.

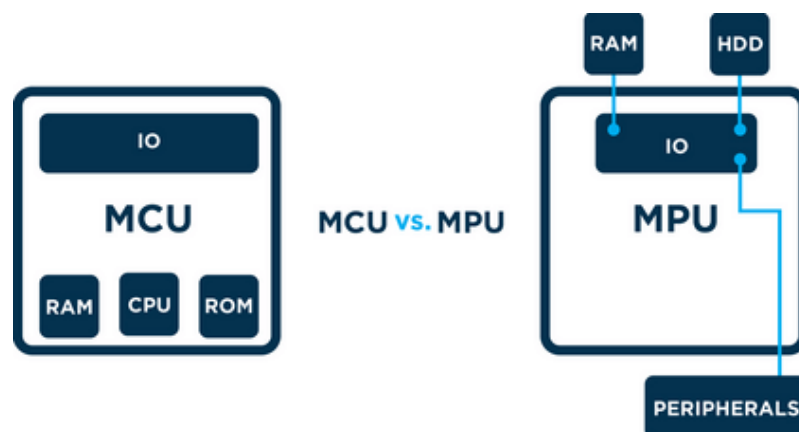


Figura 7. Diferencias entre MCU y MPU [23].

- **Comunicación inalámbrica**

En los últimos años los cables han ido desapareciendo de la electrónica de consumo. Así, actualmente un *smartphone* de gama media tiene la entrada de carga de batería y la salida de audio como únicas interfaces cableadas. El extendido uso de auriculares inalámbricos y el futuro de la carga inalámbrica, hace pensar que dentro de unos años los cables “desaparecerán”, pasando a formar parte de un nivel bajo de abstracción, donde el usuario apenas tendrá que interactuar con ellos.

Evidentemente, los cables están desapareciendo debido a la comodidad que ofrecen la multitud de tecnologías inalámbricas disponibles en dispositivos como los smartphones. Un *wearable* se debe diseñar de acuerdo a las características de la aplicación, esto incluye el protocolo de comunicación inalámbrica a implementar. Las tecnologías más utilizadas en la actualidad son:

- WiFi.
- Red móvil LTE 4G.
- Bluetooth.
- ZigBee.

Para aplicaciones que necesiten grandes tasas de datos o distancias elevadas, las mejores opciones son WiFi y 4G. Por otro lado, para implementar comunicaciones que ofrezcan una robustez y seguridad elevadas, con tasas de datos pequeñas, la mejor opción es ZigBee, que está prácticamente cerrado al ámbito de la domótica.

Sin embargo, los *wearables*, generalmente, comunican pequeñas cantidades de información (relacionados con valores de sensores normalmente), y el receptor de la información suele ser el smartphone. Por lo tanto, implementar una comunicación WiFi o LTE entre ambos dispositivos resulta un sobredimensionamiento innecesario, ya que por norma general van a estar a distancias cortas, y con tasas de datos tan pequeñas conviene optar por un protocolo que prime el ahorro en consumo. ZigBee cumple con esta última condición, pero su presencia en los smartphones es nula, por lo que tampoco sería viable. No obstante, como se ha repetido en varias ocasiones es importante no descartar nunca las diferentes alternativas, ya que para según qué aplicación puede ser mejor implementar una u otra.

Por lo tanto, Bluetooth puede ser el protocolo que mejor funcione a la hora de implementar la comunicación en un *wearable*, concretamente su versión de bajo consumo (disponible a partir de la versión 4.0), *Bluetooth Low Energy* (BLE), que ofrece una comunicación para tasas de datos no muy elevadas, con un rango de unos 10 metros, aunque el máximo teórico sea de 100 metros [24].

Bluetooth Low Energy ofrece también una serie de características que hacen que el protocolo sea de fácil comprensión y por tanto sencillo de implementar. Los agentes que intervienen pueden ser de dos tipos: periférico (o servidor) y central (o cliente).

De esta manera, un dispositivo central podrá estar conectado a varios periféricos, pero un dispositivo periférico sólo podrá comunicarse con un dispositivo a la vez.

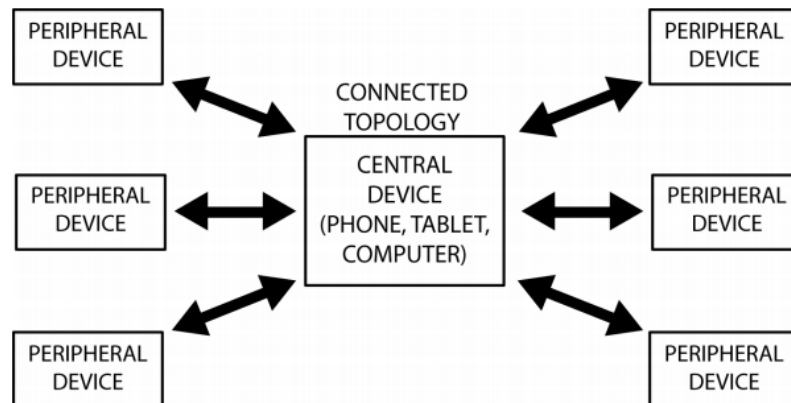


Figura 8. Topología de una red en BLE [24].

Por lo general, para que un periférico sea visible, debe estar realizando un *Advertising* cada cierto tiempo, con el fin de que un dispositivo central pueda conectarse. Cuando se establece la conexión, el dispositivo central actúa como maestro, interactuando con él o con los servicios que se han implementado en el dispositivo periférico. Estos servicios tienen la siguiente jerarquía, de acuerdo al protocolo *General Attribute Profile* (GATT):

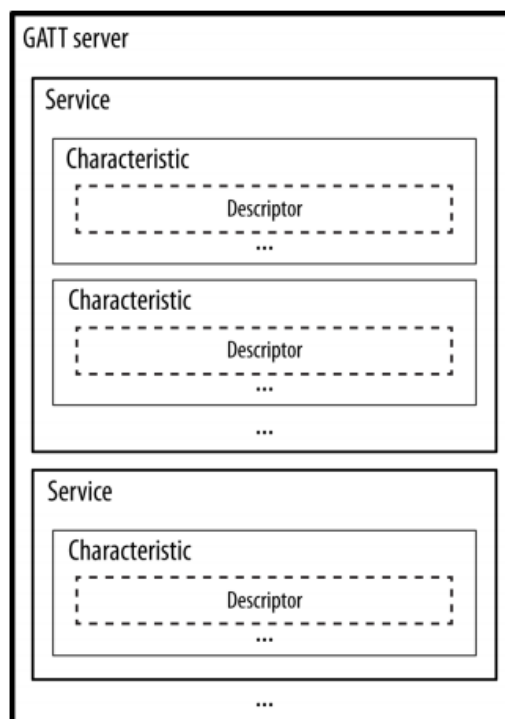


Figura 9. Esquema de servicios del protocolo GATT [24].

En este protocolo se pueden observar diferentes servicios, cada uno con una o varias características. Cada servicio se corresponde con el tipo de información que se va a gestionar por dicho canal (ritmo cardíaco, nivel de batería, etc.), y las características son los parámetros dentro de dichos servicios, a los que el usuario puede acceder en modo de escritura, lectura o notificación—similar a lectura, pero al estilo de una interrupción [24].

Cada dispositivo tiene un identificador único universal (UUID), así como cada servicio. El cliente debe conocer el UUID de los servicios que ofrece el periférico para poder establecer la comunicación.

2.1.3. *Wearables* en la atención médica

Probablemente el sector donde más aplicaciones tenga la tecnología *wearable* sea en el sector salud, de cara a monitorizar parámetros biológicos, prevenir, tratar o diagnosticar enfermedades. Ciertamente, se trata de un sector con cada vez más necesidades, ya que la población mundial, hoy de 7,38 mil millones de personas, tiene un aumento previsto hasta llegar a los 11,18 mil millones en el año 2100 (Figura 10). Esto, sumado al envejecimiento de la población [25], se traduce en más personas en los hospitales, lo que significa más volumen de trabajo para los médicos, y por ende una disminución en la calidad de la atención médica.

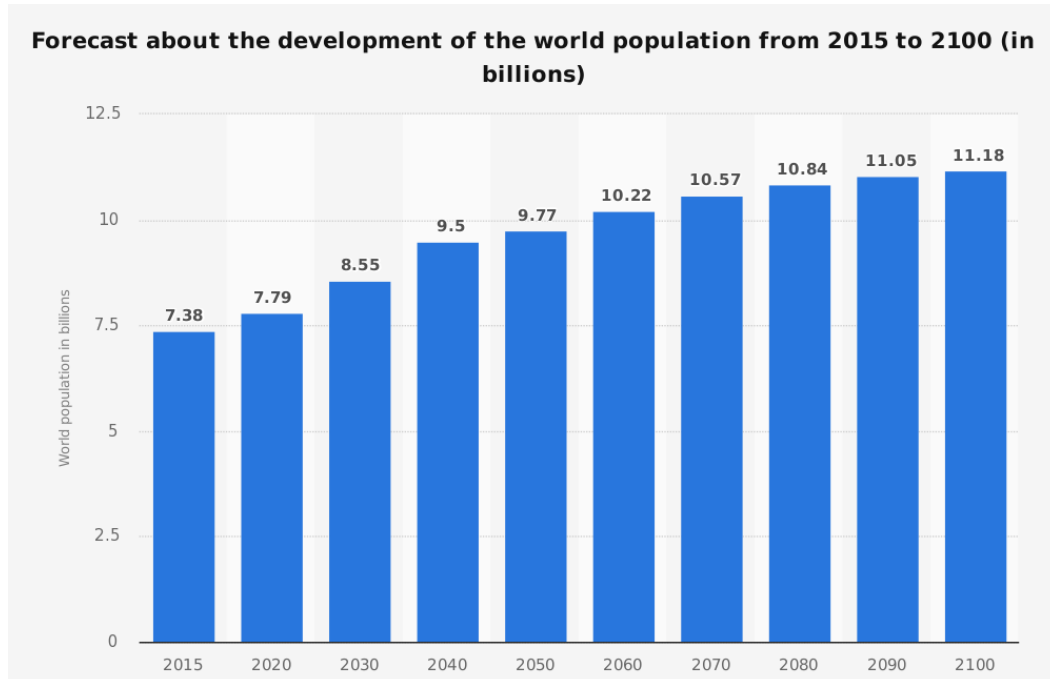


Figura 10. Crecimiento previsto de la población a lo largo del siglo [26].

La tecnología *wearable* ofrece multitud de soluciones a favor de la calidad de la atención médica, ofreciendo junto al IoT la posibilidad de monitorizar de manera continua la salud de los pacientes, posibilitando una transformación en la atención médica del futuro: de

una atención basada en el hospital, a una basada en el individuo. Por otra parte, la seguridad y salud en trabajos de cierto peligro podrían estar monitorizados constantemente [27].

Estos dispositivos pueden orientarse a cualquier etapa de la atención sanitaria. Por ejemplo, investigadores italianos han desarrollado un sistema para dar soporte a la rehabilitación tras un episodio de derrame cerebral, a través de la monitorización de los movimientos del brazo (Figura 11) [28].



Figura 11. Prenda de ropa para asistir la rehabilitación tras un derrame cerebral [28].

Por otra parte, la tecnología *wearable* puede ser una herramienta útil para reaccionar a tiempo ante una crisis. Así, se puede monitorizar a las personas de edad avanzada que viven solas y poder responder rápidamente ante una emergencia. En este contexto, los retos actuales son varios [29]:

- Accesibilidad para la población anciana. Este segmento de la sociedad es el que más veces visita al médico al año, por lo que conviene que se acostumbren a interactuar con esta tecnología. Esto supone consideraciones y especificaciones en el diseño.
- Privacidad. Si se quieren compartir los datos de los pacientes, se debe hacer con una seguridad elevada, ya que se trata de información personal y confidencial.
- Procesado de datos mediante *Big Data*. Este punto es interesante para poder automatizar y optimizar la interpretación de la información sanitaria de los pacientes.
- La aparición de sistemas de monitorización no intrusivos, ofreciendo cierto nivel de integración y de conectividad.

2.2. Técnicas de monitorización de ritmo cardíaco

En primer lugar, es importante diferenciar las dos maneras de monitorizar el ritmo cardíaco más utilizadas en la actualidad: el **electrocardiograma** (ECG) y la **fotopletismografía** (PPG). La primera de ellas se basa en la detección de pulsos eléctricos producidos por el corazón, mediante la colocación de varios electrodos en la superficie del cuerpo. Por otro lado, la PPG usa una técnica menos intrusiva pues emplea señales

ópticas emitidas por un LED, que responden a la presencia del flujo sanguíneo mediante la reflexión y absorción. Un fotodetector se encarga por tanto de obtener la señal reflejada. Para ambos casos, suele ser necesario después un procesado digital de la señal (DSP), que ayude a obtener el valor del ritmo cardíaco en unidades de pulsos por minuto (bpm) [30].

La tecnología basada en PPG se ha venido usando en los últimos años al tratarse de una técnica mínimamente invasiva. Sin embargo, se ha comprobado que cuando el ritmo cardíaco aumenta a valores elevados (en torno a 155 bpm), la correlación entre el ECG y la PPG se reduce, siendo el ECG más preciso [30]. Por lo tanto, la meta en esta tecnología es minimizar este error, manteniendo una técnica poco invasiva sin un impacto importante en la precisión del sensor.

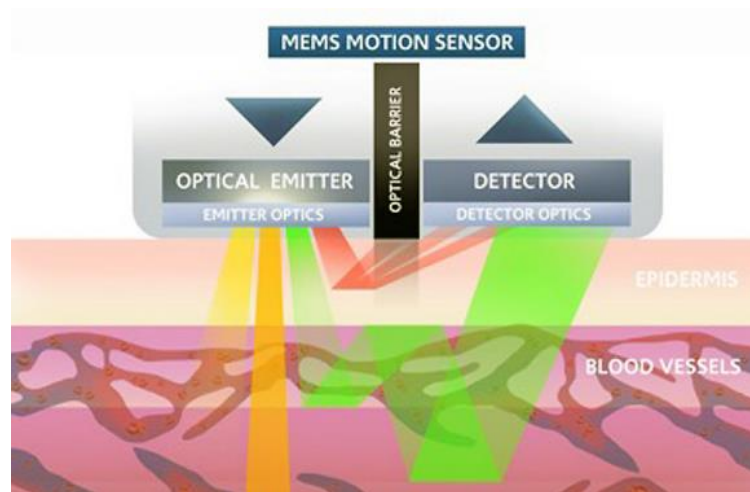


Figura 12. Esquema de funcionamiento de la técnica PPG [31].

En los últimos años, a pesar de la predominancia de estas dos técnicas a nivel comercial, han aparecido diferentes métodos de monitorización no intrusivos, que merece la pena mencionar [32]:

- **Balistocardiograma (BCG)**: detección de vibraciones mecánicas del cuerpo, aplicando diferentes tecnologías. Por el principio físico, esta técnica es la que más se asemeja a la estudiada en el presente trabajo, aunque la obtención de la información se realiza con dispositivos sensibles a las vibraciones mecánicas, como son las galgas extensiométricas. Un ejemplo de la señal obtenida se puede observar en la Figura 13 [33], en comparación con una señal de ECG. Como se puede comprobar, la señal de BCG se extiende más en el tiempo que la de ECG, por lo que la detección de los picos es por tanto más compleja.

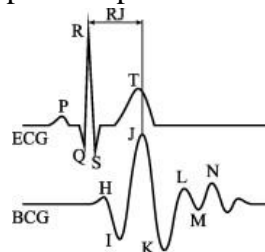


Figura 13. Señal ECG vs BCG [33].

- **Electrocardiograma capacitivo (cECG):** se basa en el cambio de la capacidad entre varios electrodos, relacionado con el paso de la sangre. Su gran ventaja es que, aún basándose en el ECG, no necesita contacto directo. Sin embargo, presenta problemas debidos al movimiento del paciente.
- **Termografía por infrarrojos (IRT):** basa sus medidas en las radiaciones infrarrojas emitidas por el cuerpo. No obstante, esta técnica requiere de calibraciones frecuentes y complejas.

Otros métodos recientes han empleado fibra óptica para monitorizar el ritmo cardíaco. Destacan sus aplicaciones para monitorización en una resonancia magnética. En la Figura 14 se puede observar el sistema propuesto por Łukasz Dziuda et al. basado en una red de Bragg sobre fibra óptica [34].

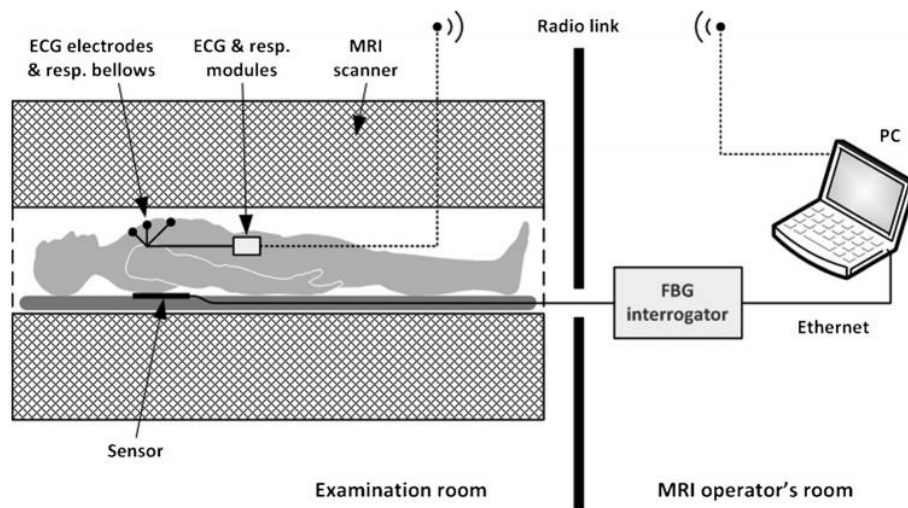


Figura 14. Sistema basado en FBG para monitorización de ritmo cardíaco [34].

2.3. Fenómeno de moteado (*speckle*)

2.3.1. Introducción. Aspectos teóricos.

El moteado o *speckle* se define como “la distribución aparentemente aleatoria que se forma cuando una fuente de luz coherente incide sobre una superficie, resultado de las interferencias constructivas y destructivas de las ondas luminosas” [35]. Este fenómeno es fácilmente visible por cualquier observador que presencie el uso de una fuente de luz coherente como la de un láser. El aspecto del patrón formado se puede ver en la Figura 15, donde se muestra un patrón con un aspecto de moteado, que resulta de la imagen final tras introducir un haz de luz coherente de un láser y hacer pasar esta luz por una fibra óptica.

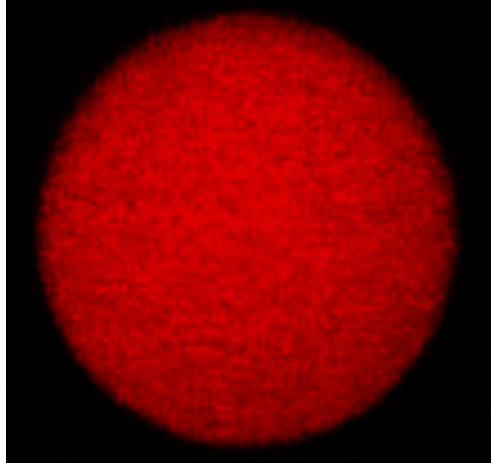


Figura 15. Patrón de speckle obtenido experimentalmente.

Como se puede observar, este moteado se compone de puntos rojos y puntos negros. Las zonas donde los puntos son negros se producen debido a la aparición de interferencias destructivas entre los diferentes modos o rayos de luz en ese punto. Por el otro lado, los puntos rojos de luz se producen al aparecer interferencias constructivas.

Al hacer pasar la luz por la fibra óptica, se presenta la capacidad de interactuar mecánicamente con la misma, de manera que cualquier cambio en el camino óptico debido a una vibración o presión ejercida sobre la fibra resulta en un cambio en el patrón de *speckle*. Este principio físico es en el que se basa el sistema del presente trabajo, proporcionando una sensibilidad importante ante vibraciones mecánicas de pequeña amplitud.

2.3.2. Aplicaciones y estado de la técnica.

Las técnicas relacionadas con el fenómeno *speckle* son varias, presentando una lista de aplicaciones aún abierta a nuevos resultados de investigaciones. Apenas unos años después de la invención del láser, este fenómeno fue utilizado por A. Labeyrie para obtener imágenes estelares [36]. A pesar del elevado interés que presenta esta aplicación, se escapa de los márgenes del presente trabajo, por lo que se prescindirá desarrollar el extendido uso que ha tenido el *speckle* en el sector de la astronomía en los últimos años, centrandó la atención en otras aplicaciones relacionadas con el ámbito o la metodología del trabajo.

La principal aplicación, y la más desarrollada en el ámbito de la biomedicina es la Imagen por Contraste de *Speckle* mediante Láser (LSCI, *Laser Speckle Contrast Imaging*), un método empleado para visualizar la corriente sanguínea, caracterizado por su buena resolución (espacial y temporal), bajo coste y simplicidad [37].

El torrente sanguíneo produce *scattering* o dispersión al iluminar con luz coherente un tejido biológico, ya sea con luz roja o cercana al infrarrojo. Esta dispersión genera un patrón de *speckle* que se puede recoger a través de la apertura de una cámara. El post

procesado de la señal obtenida produce imágenes con resoluciones elevadas, en el orden de los $10\mu\text{m}$ [38]. Este procesado consiste en la integración en el tiempo de los patrones de *speckle* obtenidos. El contraste de *speckle* se calcula como sigue:

$$K(x, y) = \frac{\sigma_N}{\mu_N} \quad (1)$$

Donde σ_N y μ_N son, respectivamente, la desviación estándar y la media de la intensidad de los píxeles en un entorno de N píxeles predefinido alrededor del píxel $P(x, y)$. Este valor calculado y la velocidad de la dispersión están relacionados de manera inversa, de forma que las zonas oscuras serán las venas transportando sangre, y las zonas claras el entorno estático. Se puede comprobar una imagen obtenida en comparación con otras técnicas de imagen clínica en la Figura 16c tomada de [38].

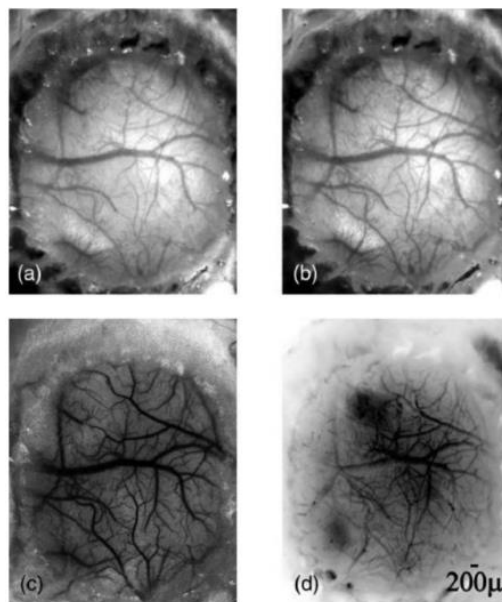


Figura 16. Comparación de diferentes técnicas de imagen médica [38].

Por otra parte, mediante una metodología similar a la que se presenta en este trabajo, el patrón de *speckle* se ha empleado para aplicaciones estructurales, posibilitando la monitorización la integridad de un edificio mediante la detección de microvibraciones mecánicas, a través de fibra óptica [39].

Por último, se va a comentar el estado de la técnica de monitorización de constantes vitales mediante el patrón de *speckle*, basando el estudio en los resultados obtenidos en diferentes investigaciones [1], [40], [41] y [42].

En cuanto a la configuración del sistema, en [1] se propone una arquitectura como la de la Figura 17. En primer lugar, un láser inyecta luz en la fibra, a través de un sistema óptico. La fibra, que es el elemento transductor, recoge las vibraciones producidas por el ritmo cardíaco del paciente y la luz se refleja en el otro extremo en un espejo que retorna la señal óptica. Es aquí cuando cobra sentido el sistema óptico de la entrada, que a la vez actúa como salida. Un *beam splitter* sensible a la polarización (PBS, *Polarizing Beam*

Splitter) refleja parte de la luz que llega de la fibra hacia una cámara comercial barata que se encarga de recoger la imagen para el post procesamiento.

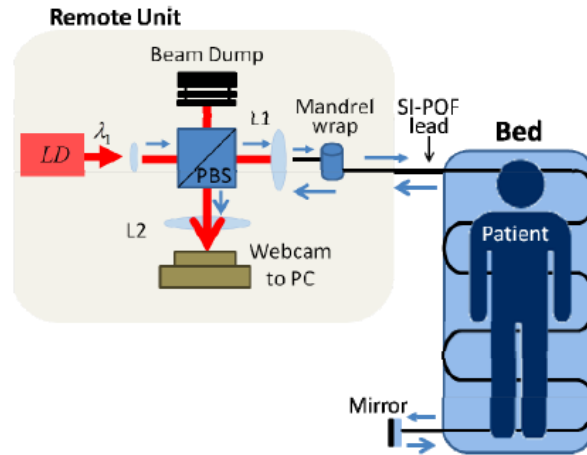


Figura 17. Sistema propuesto por P. J. Pinzón et al. en [1].

En este caso, el paciente está acostado en un colchón con la fibra óptica embebida, como ocurre con el sistema propuesto en [40] y [42]. Sin embargo, los resultados obtenidos tanto en [1] como en [40], con errores bajos en torno al 3%, invitan a ampliar la aplicación del sistema, y convertir el sistema en portátil, siendo interesante por tanto implementar un sistema de detección de movimiento por software como el que se propone en [41].

Sin embargo, existe una diferencia destacable respecto a la configuración propuesta en [42]. Estando el paciente igualmente recostado en una cama, el patrón de *speckle* se toma mediante un array óptico lineal de 102 píxeles de longitud, en vez de con una cámara, como se propone en el resto de la literatura, procesando posteriormente con un microcontrolador ATmega168. De la misma manera, la frecuencia de muestreo tomada es de 1kHz, siendo considerablemente mayor a la frecuencia de muestreo óptima para detectar el ritmo cardíaco. Aún así, el sistema es capaz de detectar tanto ritmo cardíaco como respiratorio.

En cuanto al procesamiento, los diferentes métodos convierten las imágenes captadas por la cámara en una señal unidimensional, para después realizar un análisis en frecuencia y obtener el ritmo cardíaco. Las técnicas para realizar esta conversión, sin embargo, son distintas. En [1], el método elegido es el de la correlación entre una imagen de referencia R , obtenida realizando la media de las últimas N imágenes, y la imagen actual P , solamente en su componente roja:

$$Corr(P, R) = \frac{\sum_m \sum_n (P_{mn} - \bar{P}) (R_{mn} - \bar{R})}{\sqrt{(\sum_m \sum_n (P_{mn} - \bar{P})^2) (\sum_m \sum_n (R_{mn} - \bar{R})^2)}} \quad (2)$$

Donde \bar{P} y \bar{R} son, respectivamente, la media de la imagen actual y la de referencia; m es el número de fila y n el número de columna, de tal manera que P_{mn} se corresponde con el valor en rojo del píxel en la fila m y columna n de la imagen P .

Por otra parte, en [40] se toma un método de procesamiento diferencial, calculando la variación producida entre cada imagen y la directamente anterior, definida como ΔI_D y que viene dada por:

$$\Delta I_D\{i\} = \frac{1}{K \cdot MN} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |I_{nm}^{i-1} - I_{nm}^i| \quad (3)$$

Donde K es el fondo de escala del color utilizado, de manera que se normaliza el valor final. Para una imagen monocromática de 8 bits, sería de 255. N y M representan el tamaño de la imagen en píxeles horizontales y verticales, e I_{nm}^i es el valor del píxel en la posición n, m en el instante i .

Por último, en [42] se propone la técnica del cambio de fase o *phase shift*, empleada anteriormente en el campo de la interferometría. Para ello, utilizando un sensor lineal o *array*, con una única fila de píxeles, se toman tres imágenes consecutivas de un vídeo, I_n , I_{n+1} , I_{n+2} y se computan sus píxeles p_n , p_{n+1} , p_{n+2} para calcular el cambio de fase, θ_n , referido a un solo píxel x , como sigue:

$$\theta_n(x) = \tan^{-1} \left[\sqrt{3} \frac{p_{n+2}(x) - p_{n+1}(x)}{2p_n(x) - p_{n+1}(x) - p_{n+2}(x)} \right] \quad (4)$$

Tras calcular este valor para cada píxel, se convierte la señal en unidimensional sumando los valores del cambio de fase calculados en Eq. (4), obteniendo un valor de señal $s(n)$ para cada imagen:

$$s(n) = \sum_{x=0}^{L-1} \theta_n(x), \quad 0 \leq n \leq N-3 \quad (5)$$

Las limitaciones actuales se presentan principalmente en los errores introducidos por los movimientos del paciente, así como en la convergencia en cuanto a un sistema de procesamiento lo suficientemente robusto como para detectar el pulso en situaciones de contingencia, además de posibilitar esta detección en tiempo real.

Capítulo 3

Sistema completo

En este capítulo se pretende ofrecer una visión a alto nivel del sistema desarrollado, con el fin de que el lector pueda diferenciar correctamente los distintos bloques y subsistemas que lo componen. En primer lugar, se hablará del sistema previamente existente, comentando las cuestiones más relevantes del mismo, para dar paso después a una introducción general del diseño realizado en el presente trabajo.

3.1. Sistema previo

Antes de comenzar a describir el diseño que se ha llevado a cabo en este trabajo, resulta preciso describir el sistema existente previamente, sobre el que se realizó un primer trabajo de aprendizaje sobre el concepto que rodea al presente proyecto. Se ha decidido dividir la explicación de lo que había implementado en dos partes, tal y como se ha dividido el diseño: *hardware* y *software*.

Este primer sistema forma parte de trabajos posteriores a la publicación ya mencionada de P. J. Pinzón et al. [1], tras los cuales se realiza un análisis llevado a cabo por Ignacio Sánchez et al. [43] con el fin de caracterizar los errores en las medidas y poder extraer conclusiones en cuanto a hardware y software.

Aunque se han analizado varias arquitecturas, en la Figura 18 se puede ver la más compleja por número de componentes, de todas ellas. En esta figura se puede observar cómo la luz de un láser (LD) se acopla a la fibra pasando antes por un polarizador ajustable y un divisor de haz sensible a la polarización (PBS). Así, sólo parte de esta luz se acopla en la fibra, donde recorrería el chaleco y se reflejaría en el otro extremo en un espejo, para volver a pasar por el chaleco, realizando el mismo camino a la inversa. Finalmente, el PBS reflejaría de nuevo parte de la luz, direccionándola hacia una cámara de bajo coste con conexión USB, concretamente la cámara PS3 Eye de Sony.

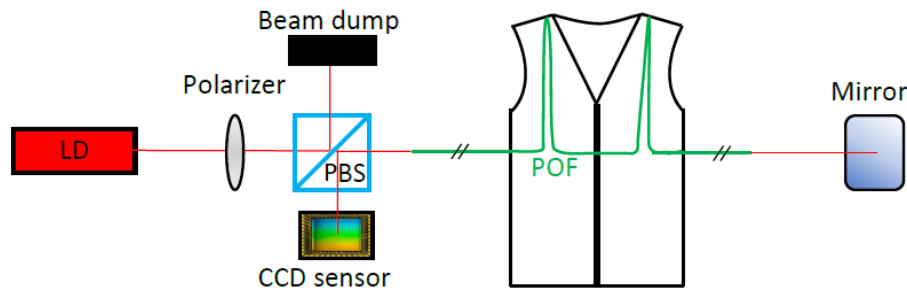


Figura 18. Configuración más compleja del sistema previo.

A lo largo del trabajo previo, se fueron retirando primero los elementos que aumentan el coste del sistema, como son el polarizador y el PBS, con el fin de extraer conclusiones sobre la influencia de efectos como la polarización en las medidas, para finalmente poder proponer un sistema como el presentado en la Figura 19, donde la luz del láser se acoplaría directamente a la fibra, donde pasaría por el chaleco e incidiría directamente en la cámara. La ventaja de un sistema donde se ha simplificado tanto la óptica resulta evidente cuando la intención final es la de integrar todo el hardware en un espacio reducido y, sobre todo, hacerlo portátil.

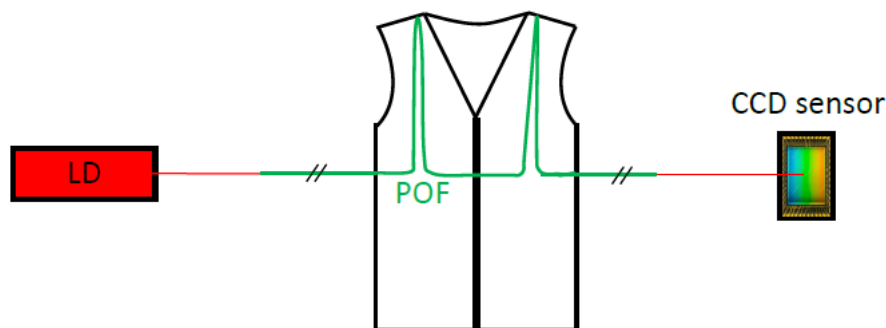


Figura 19. Configuración más sencilla del sistema previo.

En cuanto al procesado, es importante destacar el procedimiento realizado. Al no existir *drivers* que permitieran utilizar la cámara en tiempo real mediante MATLAB, se realizaban grabaciones de 2 minutos, ajustando antes los diferentes parámetros de configuración de la cámara para poder observar el *speckle* mediante un software propietario dedicado a dicha cámara. Por lo tanto, se realizaba posteriormente un procesado *offline* mediante MATLAB en un ordenador, utilizando el mismo software de procesamiento propuesto en [1] por P. J. Pinzón et al.

Este procesado, como se ha comentado en el epígrafe 2.3.2 utiliza el método de la correlación entre cada imagen y una imagen de referencia resultante de una media móvil entre todas las imágenes anteriores, todas ellas en su componente roja. Tras esto, se obtiene una señal unidimensional que se filtra y se procesa mediante diferentes técnicas de análisis espectral.

Alguna de las señales obtenidas se pueden ver en la Figura 20, donde en la parte superior se encuentra la señal de correlación, tras hacerla pasar por una función *detrend* que elimina cualquier tendencia lineal. En la parte inferior, se muestra la señal filtrada mediante un filtro FIR pasa banda con un ancho de banda entre 0,6 y 1,8 Hz.

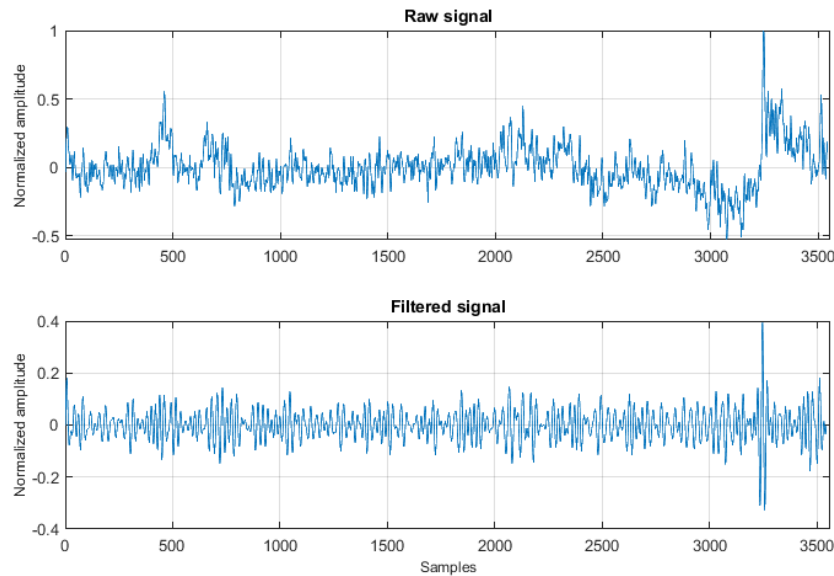


Figura 20. Señales obtenidas con el sistema previo.

Tras obtener la señal filtrada, la aplicación del espectrograma resultaba en una gráfica como la de la Figura 21. Como se observa, las oscilaciones entre unas frecuencias y otras son bastante pronunciadas (línea negra), por lo que se ha implementado cierta inteligencia en el software con el fin de suavizar estas oscilaciones (línea roja).

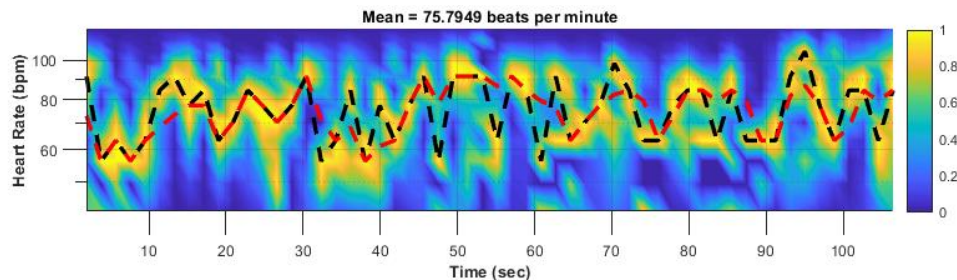


Figura 21. Análisis espectral de una grabación con el sistema previo.

Los principales cambios que se van a realizar en el presente trabajo con respecto al sistema previo serán de diseño de un *hardware* específico y su integración para hacer el sistema portátil, incluyendo el uso de una placa basada en microprocesador para poder realizar el procesamiento de las imágenes. Además, se realizará también una integración de *software*, valorando el uso de diferentes algoritmos de procesamiento y de comunicación, teniendo en cuenta que el carácter del procesamiento pasará a ser en tiempo real. Además, se añadirá una aplicación en Android para el control inalámbrico del sistema.

3.2. Diseño del sistema portátil

Como ya se ha comentado previamente, la filosofía del diseño realizado es la de simplificar al máximo el tamaño del *hardware* del sistema. Esta simplificación se ha realizado sobre todo en la óptica, llegando a realizar un acoplo directo tanto en la transmisión como en la recepción de la luz. Por otra parte, se ha realizado una simplificación en la electrónica necesaria para que funcione el sistema, utilizando una placa de microprocesador alimentada mediante una batería. El sistema implementa también una comunicación *Bluetooth* que hace posible tomar medidas de manera inalámbrica.

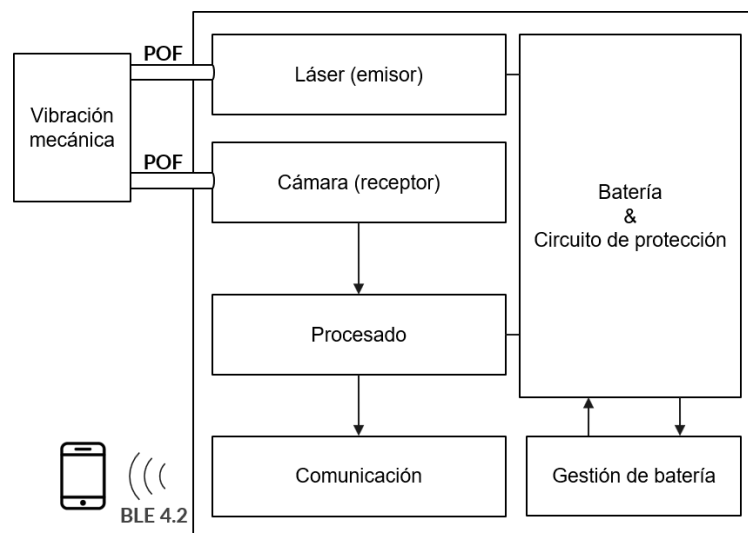


Figura 22. Arquitectura general del sistema.

El diagrama de bloques del sistema completo se observa en la Figura 22. De forma específica, se usa fibra óptica de plástico (POF) como medio de propagación de las vibraciones, de tal manera que podrá estar embebida en el chaleco como se ha visto en el epígrafe anterior. En este trabajo la fibra estará en contacto con un zumbador electrónico que se utilizará para depurar el procesamiento y como emulador de la aplicación final. Por otro lado, en un extremo de la fibra se encuentra el diodo láser, encargado de proporcionar la luz coherente que posibilita el efecto *speckle* en el que se basa el sistema. En el extremo contrario, debidamente acoplado a la salida de la fibra, se tiene un receptor que recoge este patrón de *speckle*. En forma de vídeo, esta información se transmite al procesador, que se encargará de realizar el procesamiento de las imágenes y calcular el ritmo cardíaco del sujeto o la frecuencia de la vibración inducida en tiempo real.

Precisamente en el procesamiento de las imágenes es donde también se realiza una búsqueda de algoritmos que puedan funcionar para monitorizar las vibraciones inducidas en la fibra y, por tanto, visibles en el patrón de *speckle*. Estos algoritmos intentarán analizar las variaciones entre las imágenes que se van obteniendo mediante la cámara, para después poder realizar un análisis espectral de la señal.

Las especificaciones de diseño del sistema completo, por lo tanto, son las siguientes:

- Se deberá optimizar el consumo global al tratarse de un sistema alimentado por batería, lo que influirá en la elección de varios elementos como el procesador, la propia batería o la tecnología de comunicación inalámbrica.
- La cámara deberá ser del menor tamaño posible y de fácil manejo para una correcta integración.
- Al ser un dispositivo alimentado mediante una batería, será preciso disponer de la capacidad de monitorizar la energía restante de la batería.
- El procesamiento de datos debe encontrar un equilibrio entre eficacia y eficiencia, que estará limitado por la velocidad de reloj del microprocesador, al tener que implementar un sistema de procesado en tiempo real.
- El láser deberá tener una longitud de coherencia lo suficientemente grande para permitir observar el *speckle* tras recorrer la longitud de la fibra.
- El sistema completo deberá poder cumplir con la normativa aplicable.

El diseño específico de cada uno de los bloques se presentará en los siguientes capítulos, en el capítulo 3 el diseño del hardware y en el capítulo 4 el diseño del software.

La integración final incluirá las medidas que permitan validar el sistema en su conjunto, así como la funcionalidad del mismo. Para tomar las medidas y poder entre otros depurar el código, en este trabajo se va a emplear un zumbador electrónico como dispositivo de ensayo (DUT), que estará en contacto con la fibra óptica. Este zumbador se podrá modular mediante un generador de funciones que permita excitarlo con la frecuencia deseada, lo que será útil de cara a caracterizar los algoritmos en términos de rendimientos según qué frecuencias. Esto se hará evidentemente con frecuencias entre 0,75 Hz y 3 Hz, de acuerdo a los límites aceptables de frecuencia cardíaca encontrados en la mayoría de las personas (45 bpm – 180 bpm).

Capítulo 4

Diseño del hardware

En el presente capítulo se tratará de justificar las decisiones tomadas en referencia a los distintos bloques *hardware* del sistema. En primer lugar, se compararán diferentes alternativas de procesadores en el mercado para después escoger una de ellas de acuerdo a las necesidades. Tras esto, se pasará a comentar la parte optoelectrónica del sistema. Por otra parte, se justificarán diferentes decisiones de diseño tomadas tanto para la gestión del consumo como para la lógica de funcionamiento del sistema, en el marco del diseño electrónico. Por último, se realiza una propuesta de integración del sistema dentro de una carcasa 3D, con el fin de poder cerrar el diseño físico del mismo.

4.1. Diseño del procesador

En este apartado hay una gran cantidad de alternativas, por lo que a continuación se van a exponer las especificaciones con mayor peso en la elección de este elemento, para posteriormente poder tomar la decisión de diseño.

En primer lugar, el **tiempo de desarrollo** es crucial en este aspecto, por lo que se debe intentar no desviar la atención del resto de elementos a diseñar. Disponer de una placa de desarrollo ya implementada, con unas herramientas de desarrollo intuitivas, se convierte en un claro punto a favor.

El hecho de que el sistema a implementar sea un *wearable*, añade tres factores importantes en el diseño. Por una parte, el diseño final debe ser del menor **tamaño** posible, para que no moleste al sujeto que lo porta. Por otro lado, el **consumo** de energía juega también un papel importante, pues el sistema estará alimentado por una batería. Además, debe ser un equipo que preferiblemente tenga capacidad de **comunicarse** de manera inalámbrica, por lo que habría que buscar sistemas que soporten Bluetooth (a ser posible BLE) y/o WiFi, además de poder almacenar los datos necesarios.

Por otra parte, atendiendo a la aplicación, la elección del procesador depende en gran manera de la cámara escogida. La transmisión y el procesamiento de las imágenes es una de las tareas que demandan más recursos de un sistema, por lo que el procesador en cuestión debe contar con la suficiente memoria RAM (del inglés *Random Access Memory*) que permita al software realizar las operaciones necesarias. Esta limitación no ha aparecido antes debido a que las pruebas se han realizado en ordenadores con 8GB de RAM, pero al tener que utilizar ahora equipos mucho más limitados, conviene tener muy en cuenta este factor.

Durante el desarrollo del software inicial en MATLAB, los vídeos se grababan y se procesaban *a posteriori*, pero se puede comprobar el tamaño de la variable que se utilizaba como referencia, compuesta por 50 *frames* de vídeo, para hacerse una idea de las necesidades de memoria RAM en el caso de que se quiera realizar un procesamiento en tiempo real. El tamaño sería el siguiente:

$$320 \cdot 240 \frac{\text{píxeles}}{\text{frame}} \cdot 4 \frac{\text{bytes}}{\text{píxel}} \cdot 50 \text{ frames} = 14.65\text{MB} \quad (6)$$

Este valor demuestra que los equipos basados en microcontrolador con valores relativamente bajos de memoria RAM, resultarían bastante limitados para la aplicación en cuestión. Anteriormente se ha comentado como con una placa Arduino (ATmega168) se han logrado realizar medidas del ritmo cardíaco mediante el fenómeno de *speckle*, utilizando en este caso el array TLS3301CL de 102 píxeles, funcionando a 1KHz [42]. Si se realiza el mismo cálculo anterior, teniendo en cuenta que la información aquí se da en valores de 1 byte (8 bits), nos queda la siguiente memoria utilizada:

$$102 \cdot 1 \frac{\text{píxeles}}{\text{frame}} \cdot 1 \frac{\text{byte}}{\text{píxel}} \cdot 50 \text{ frames} = 5.1\text{KB}, \quad (7)$$

memoria que sí es asumible por un sistema basado en microcontrolador. Por esto, si se va a utilizar un equipo de estas características, nos encontraríamos forzados a elegir el array de fotodetectores como elemento receptor.

De aquí en adelante, con el fin de proponer equipos lo más eficaces posibles, se tomará la decisión de cara a poder continuar utilizando una cámara, limitando por encima las necesidades de procesamiento que se necesitarían para el uso de un array de fotodetectores.

Para repasar los productos disponibles en el mercado, se han dividido los mismos en tres grupos, buscando después diferentes alternativas que ofrecen los fabricantes para cada caso. El primero de los grupos son los **SBC** (del inglés *Single Board Computer*), cuyo nombre las define adecuadamente; son placas de pequeño tamaño con un procesador integrado, y con la suficiente capacidad para correr sistemas operativos, normalmente basados en Linux. En muchos casos se dispone de múltiples interfaces de comunicación,

tanto por cable como inalámbricas, y pueden llegar a estar disponibles en tamaños muy reducidos, que es lo que se busca en este caso. El hecho de que se disponga de un sistema operativo supone una libertad mayor a la hora de programar el software necesario, además de un rendimiento importante. Sin embargo, estas placas presentan un mayor consumo de energía que otras alternativas, debido principalmente a la potencia del microprocesador y al consumo por parte de las diferentes interfaces disponibles.

En el segundo grupo se encuentran los **MCU y MPU de bajo consumo**. Estos dispositivos se caracterizan principalmente por tener un manejo eficiente de la energía. Evidentemente, esto se consigue a costa de reducir el rendimiento, teniendo frecuencias de reloj mucho más bajas que el primer grupo. Además, la memoria RAM es bastante limitada, sin apenas capacidad para manejar los datos de la cámara calculados anteriormente. En algunos casos, existe la posibilidad de ampliar la memoria RAM, así como de disponer de placas de evaluación para los dispositivos. Sin embargo, estas placas suelen ser de un tamaño elevado, haciéndolas poco prácticas para la aplicación. En el caso de no disponer de placa de evaluación, se entraría en un complejo proceso de diseño de una placa para el MCU/MPU que, como se ha comentado anteriormente, tendría un peso importante en el tiempo de desarrollo. A la hora de comparar componentes de este grupo, se ha decidido partir de familias de MCUs/MPUs de bajo consumo de los diferentes fabricantes, tomando como referencia el dispositivo de cada familia con mayor rendimiento y memoria RAM, ya que el consumo es un apartado en el que cumplen todos.

Por último, en un tercer grupo se han introducido procesadores digitales de señal (**DSPs**, del inglés *Digital Signal Processors*) que resultan dispositivos interesantes a tener en cuenta para la aplicación. Como en el grupo anterior, la memoria RAM es escasa en general, pero se puede ampliar en algunos casos. Por otra parte, las interfaces disponibles son bastante pocas, y su implementación dependería, como en el caso anterior, del diseño de una PCB (del inglés *Printed Circuit Board*) o placa de circuito impreso, con las propiedades necesarias para el funcionamiento del DSP. El espacio de tiempo que ocuparía el proceso de diseño y depuración de una placa para estos dispositivos desplazaría otras tareas más importantes en el presente proyecto, como el desarrollo y la puesta a punto del software. No obstante, esta opción es óptima si se tuviera el tiempo suficiente para llevar a cabo el diseño completo, así como la familiaridad en el diseño de PCBs para alta velocidad. El acercamiento en este grupo funciona de la misma manera que el grupo anterior, teniendo en este caso menos ejemplos de los diferentes fabricantes.

Los dispositivos escogidos se pueden ver en la Tabla 2, con sus respectivos grupos. Se ha realizado un recorrido por los diferentes fabricantes, y se han escogido las familias de productos que mejor se pudiesen adaptar a la aplicación a desarrollar en el presente trabajo.

Tabla 2. Lista de productos considerados para el diseño del procesador.

Tipo de producto	Familia/Producto	Fabricante
SBC	Raspberry Pi Zero W	Raspberry Pi
	Banana Pi Zero	Banana Pi
	PocketBeagle	Circuitco LLC
	Carambola 2	8devices
	LinkIt Smart 7688 Duo	Seeed Studio
	Onion Omega 2	Onion
	NanoPi NEO Air	FriendlyARM
MCUs/MPUs de bajo consumo	SAMA5D3	Microchip
	STM32L5	ST Microelectronics
	STM32L4	
	PSoC	Cypress
	MCF5301x	NXP
	MSP430	Texas Instruments
DSPs	dsPIC33E	Microchip
	TMS320	Texas Instruments

Con estos dispositivos, se ha realizado un estudio completo, analizando las diferentes propiedades de cada uno y comparando cada propiedad, obteniendo una comparación general que puede ayudar a tomar la decisión de diseño. La comparativa se puede consultar en el Anexo III. Comparativa procesadores.

Los precios de cada producto se han tomado, en su mayoría, de las tiendas de Digi-Key, RS, Mouser y Farnell. Para presentar las características de cada familia de productos, se ha tomado un producto de referencia, tal y como se muestra en la Tabla 3.

Tabla 3. Familias y productos de referencia de los procesadores a comparar.

Familia	Producto referencia
SAMA5D3	ATSAMA5D36
STM32L5	STM32L552xx
STM32L4	STM32L496xx
PSoC	PSoC 63 BLE
MCF5301x	MCF5301x
MSP430	MSP430F5255
dsPIC33E	dsPIC33EP512GP806
TMS320	TMSC5505

De esta comparativa, se puede extraer finalmente una conclusión, resultando la Raspberry Pi Zero W por encima del resto en muchos aspectos, sobre todo superando con solvencia

la capacidad de procesamiento necesaria, sin presentar un consumo extremadamente elevado. Por otra parte, la proactiva comunidad de la que disfruta este dispositivo es también un punto a favor para su elección. La alternativa de emplear un dispositivo lineal como el TMS320C5505, parece también una buena opción de cara a un diseño definitivo, donde habría que desarrollar y depurar una PCB basada en este dispositivo, incluyendo la expansión de memoria RAM. Como en este caso la materia de estudio responde a diferentes aspectos, se optará por una plataforma con la que el trabajo sea lo más fluido posible. En este trabajo se intentará diseñar en su versión inicial una demostración de un *wearable* utilizando la tecnología presentada en el estado del arte, y el extendido uso de la **Raspberry Pi Zero W** en el IoT permite un desarrollo cómodo de este paso intermedio hacia el desarrollo completo de un *wearable*.

El principal problema que surge de la elección de la Raspberry Pi es la elección de una batería que pueda asegurar una autonomía suficiente al sistema, sin suponer un aumento considerable del tamaño de éste. Este aspecto se tratará en detalle en el apartado de Batería y gestión energética

4.2. Optoelectrónica

4.2.1. Fibra óptica

Analizando diferentes estudios realizados para el sensado mediante fibra óptica de microvibraciones [39], [1], [40], [41] y [42] se observa que coinciden en que la utilización de fibra óptica de plástico (POF) es una buena opción. En concreto, las principales ventajas del uso de POF para esta aplicación son varias:

- Las fibras ópticas de plástico presentan una buena biocompatibilidad que las hace adecuadas para su uso en medicina [44].
- La elasticidad que presentan las hace presentar una resistencia al estrés por encima del resto de fibras disponibles en el mercado. Además, son bastante dúctiles [45]. Esto se relaciona directamente con la aplicación, pues el sistema debe permitir cierto grado de flexibilidad.
- Las propias fibras y la instrumentación empleada para las mismas son de bajo coste en comparación con otras tecnologías [46].
- Tienen una apertura numérica elevada en comparación con las fibras de silicio. Esto significa que el ángulo de aceptación de la fibra será mayor, de manera que el acoplo entre el emisor y la fibra y entre fibras es más sencillo de realizar.

La principal desventaja de la POF en comparación con las fibras de silicio son sus elevadas pérdidas por unidad de longitud, típicamente de 0.2dB/m a 650nm frente a los 0.22dB/km a 1550nm de las fibras de silicio. Sin embargo, este factor no afecta al presente sistema, ya que las longitudes de transmisión son cortas.

En cuanto al perfil del índice de refracción de la fibra y su geometría, resulta necesario usar una fibra multimodo, de tal manera que haya suficientes modos para que tenga lugar el patrón de *speckle*.

Por lo tanto, la fibra a utilizar en el sistema será una fibra óptica de plástico multimodo de salto de índice (SI-POF), siendo el material del núcleo PMMA (polimetilmetacrilato).

4.2.2. Láser

Antes de comenzar a definir las condiciones que debe cumplir un láser para la presente aplicación, resulta preciso definir ciertas propiedades de éste.

En primer lugar, la principal propiedad que interesa comentar, y que atañe directamente a la aplicación que aquí se le quiere dar, es la coherencia de la luz del láser. Para definir esta característica, es importante comenzar por la generación de la luz en un láser. Es importante remarcar desde aquí que las definiciones se referirán en todo momento a tecnologías de láser basadas en materiales semiconductores.

La emisión espontánea se produce cuando en una sección de un material dado, un electrón se excita hasta llegar a un nivel de energía elevado, y automáticamente realiza una transición a un nivel más bajo, emitiendo en este proceso un fotón (Figura 23, izquierda). Este proceso es el principio de funcionamiento de los diodos LED (*Light-Emitting Diodes*). Las propiedades de la luz emitida son altamente probabilísticas, sin presentar ninguna relación en fase entre los fotones emitidos. Esto genera un patrón de emisión poco direccional y poco coherente, además de tener un ancho espectral bastante grande en comparación con otras fuentes de luz.

Por otro lado, la emisión estimulada tiene lugar en un medio activo, donde un electrón que desciende desde un nivel de energía elevado a uno más bajo reacciona con un fotón presente en dicho medio, generándose dos fotones con la misma fase y dirección (Figura 23, derecha). De esta manera, en este medio activo se produce un efecto de crecimiento exponencial donde aumenta el número de fotones con la misma fase y dirección. La emisión de luz resultante tendrá las características ya comentadas, con alta direccionalidad y casi monocromática, además de presentar una alta correlación en fase entre los diferentes fotones tanto temporal como espacial. A esta última propiedad de la luz se le llama coherencia, y es lo que posibilita el uso de las interferencias entre los distintos rayos de luz de una fuente como el láser (acrónimo de *Light Amplification by "Stimulated Emission" Radiation*). Por tanto, será primordial utilizar una fuente de luz con alta coherencia, como puede ser un diodo láser.

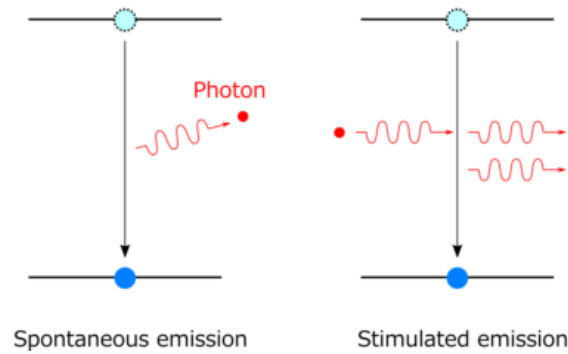


Figura 23. Emisión espontánea y emisión estimulada [47].

En cuanto al láser a emplear, interesa que cumpla las siguientes condiciones adicionales:

- La longitud de onda central debe corresponderse con la zona espectral donde las pérdidas de la fibra óptica de plástico sean menores, por ejemplo, a 650nm (color rojo).
- El ancho espectral no será crítico pues el fuerte carácter multimodal de la fibra POF por su geometría permitirá la aparición de distintos modos propagándose por la fibra, aún con un espectro reducido.
- La potencia de luz emitida no debe saturar la imagen de la cámara. Debido a la corta longitud de la fibra a emplear, se hace necesario que sea un láser de potencia reducida. De esto se deriva que reducir las pérdidas de acoplo a la fibra no será relevante, siempre y cuando el patrón de *speckle* no se vea afectado.
- Eficiencia en la conversión eléctrico-óptica y bajas corrientes de polarización que permitan un consumo lo más reducido posible del láser.
- Fácil acoplo con la fibra óptica que permita evitar la propagación en el aire que lleve a tener que tomar precauciones de seguridad adicionales en el uso del sistema, y las pérdidas que puedan redundar en un aumento del consumo.
- La longitud de coherencia del láser es otro aspecto importante a tener en cuenta para el diseño del sistema. Este parámetro de un láser, directamente relacionado con el tiempo de coherencia, cuantifica la distancia a partir de la cual se pierde la coherencia del láser. Esta pérdida de coherencia, en lo que al presente trabajo se refiere, se traduce en la imposibilidad de observar el patrón de *speckle*. Su valor viene dado por la siguiente expresión:

$$L_{coherencia} = \frac{\lambda^2}{n \cdot \Delta\lambda} \quad (8)$$

donde n es el índice de refracción, típicamente 1.49 para POFs, λ la longitud de onda central y $\Delta\lambda$ la anchura a media altura (FWHM, *Full Width at Half Maximum*) del láser.

Por tanto, teniendo en cuenta las características mencionadas, el láser a utilizar en el diseño debe ser un láser directamente acoplado a la fibra, de tal manera que la luz se encuentre siempre confinada dentro, sin que se propague luz por el aire que pudiese

incidir sobre una persona. No obstante, el sistema estará igualmente dentro de una caja que pueda evitar cualquier fuga de luz de manera redundante.

El láser inicialmente seleccionado es el modelo LOC-RM/650/FC0.5 de LasIRvis, directamente acoplado a fibra POF que se puede observar en la Figura 24. Emite luz en torno a los 650 nm, y es de relativamente baja potencia. Se ha caracterizado que emite una potencia de salida tras la fibra en torno a 1mW, medida con el medidor de potencia de Thorlabs PM20A [48]. Tiene un consumo de aproximadamente 20mA que resulta aceptable para un elemento de un sistema portátil. Por lo tanto, el posterior diseño de integración física del sistema completo se realiza con este láser.

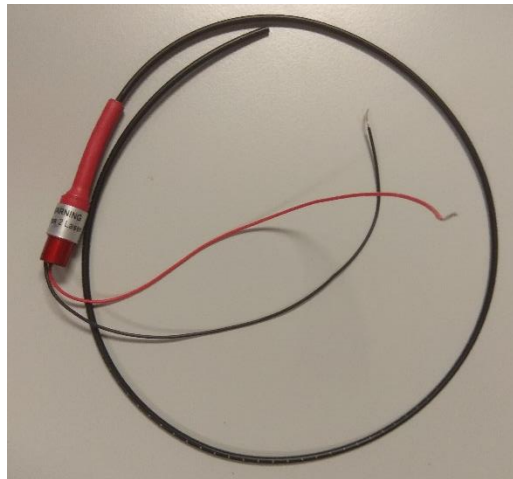


Figura 24. LOC-RM/650/FC0.5 de LasIRvis.

Sin embargo, se utilizarán también otros módulos láser de distintas características como alternativas posibles, dada la dificultad para poder disponer de vendedores alternativos del mismo.

Entre las otras alternativas, la primera es un módulo de uso como puntero láser que no está directamente acoplado a la fibra, por lo que habría que asegurar este acoplo si se utilizara en un futuro. El láser en cuestión, véase la Figura 25, es el mismo que se empleaba previamente en [1] y [43]. Este puntero, a pesar de no disponer de ninguna referencia de fabricante (más allá de su número de inventariado) había demostrado previamente su funcionamiento. El consumo, no obstante, es más elevado, en torno a los 40mA.

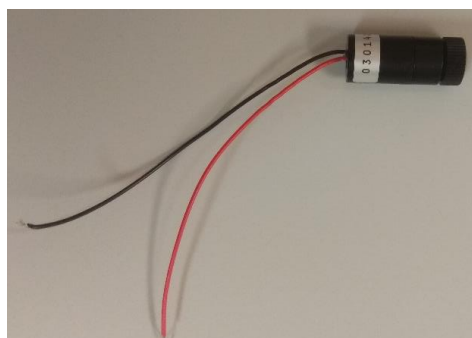


Figura 25. Láser utilizado en [1] y [43].

La segunda alternativa es otro módulo láser, cuya referencia es Akozon65Z2Z7BUTW, con longitud de onda central de 650nm y potencia óptica de salida de 5mW [49]. Este módulo consume en torno a 30mA alimentado a 5V. Se podría cambiar por tanto la resistencia en serie que posee para poder operar el láser a la misma corriente pero utilizando 3,3V, voltaje que permite un menor consumo y que es aquella con el que trabajan algunos procesadores de bajo consumo como la Raspberry Pi. En la Figura 26 se puede observar el módulo adquirido.

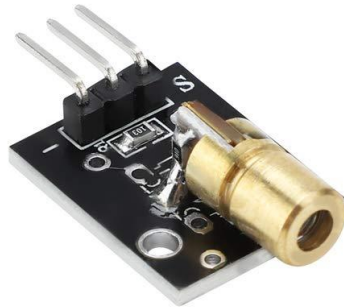


Figura 26. Láser de Akozon comprado como alternativa.

Para garantizar el correcto acoplo directo a la fibra, se decide unir dicho láser a una POF utilizando pegamento termofusible, tubos termoretráctiles y cinta aislante, quedando un dispositivo como el mostrado en la Figura 27.



Figura 27. Láser acoplado a la fibra.

4.2.3. Cámara

A la hora de elegir un sensor que permita identificar el patrón de *speckle*, aparecen tres grupos de dispositivos con diferencias importantes: las cámaras comerciales o webcams (que es la que existe en el sistema previo), los sensores CMOS o CCD, y los *arrays* de fotodetectores.

- **Cámaras comerciales**

En este caso, se pueden dar ejemplos de cámaras baratas que, como la utilizada en el trabajo previo (PS3 Eye), disponen de algunos programas que hacen posible su manipulación. Incluso, se podría intentar buscar cámaras que pudiesen ser utilizadas con MATLAB y poder realizar el análisis en tiempo real (algo que con la PS3 Eye

no es posible). En su gran mayoría, estas cámaras tienen integrada una interfaz USB para comunicarlas con un PC. Ejemplos de estas cámaras se pueden encontrar muy fácilmente, todas con características similares a las de la PS3 Eye, y con precios que pueden ir desde los unos pocos euros hasta los miles de euros en el caso de cámaras profesionales.

Las ventajas de estas cámaras las hacen elementos ideales para el trabajo que se ha realizado hasta ahora, que es el de realizar pruebas de manera estática. Son cámaras con una interfaz directa para trabajar con un PC, usando un software propietario sencillo de utilizar para poder ajustar sus parámetros.

Sin embargo, de cara a implementar un producto, se hace patente su falta de modularidad a la hora de comunicar por ejemplo con un microprocesador. Además, su tamaño tampoco es el perfecto para poder desarrollar un *wearable*. Por otra parte, aunque se pueda modificar, la resolución de estas cámaras suele ser mejor de la que realmente se necesita para detectar el patrón de *speckle* [49]¹, lo que puede significar un aumento de consumo y de necesidades de procesamiento que pueden no ser asumibles por un microprocesador.

Una excepción que puede ser incluida en este grupo, es la cámara de Adafruit [50], cuyo pequeño tamaño y extendido uso con plataformas como Arduino, ha hecho que sea una buena opción para este trabajo. El hecho de que haya cierta comunidad detrás presenta un punto a favor para esta cámara.

Aparece, por otra parte, teniendo en cuenta la posible utilización de un sistema basado en Raspberry Pi, la propia cámara de este dispositivo [51], que de hecho tiene un puerto *Camera Serial Interface* (CSI) adaptado a la propia Raspberry Pi.

- **Sensores CMOS o CCD**

Este grupo de detectores se corresponden con cámaras al nivel más bajo posible, con necesidad de acondicionamiento óptico y electrónico. Las ventajas son evidentes, ya que son totalmente modulares y de pequeño tamaño, se pueden adaptar a cualquier sistema.

No obstante, el hecho de que se trate de cámaras a un nivel muy bajo supone también una desventaja importante, ya que requiere de minuciosos ajustes electrónicos y de software en el microprocesador, lo que podría suponer tiempos de desarrollo largos. Además, como pasa con muchas de las cámaras comerciales, la resolución de estas cámaras puede llegar a ser demasiado alta, lo que supone problemas en cuanto a consumo y disponibilidad de recursos, como se ha comentado anteriormente.

¹ A modo informativo, el tamaño recomendado de los *speckles* es decir, de los puntos del moteado, está entre 3-5 veces el del pixel, para poder evitar problemas de ruido blanco.

- **Arrays de fotodetectores**

Estos elementos presentan una alternativa muy interesante, ya que se trata de formaciones lineales de fotodetectores, con un número no muy elevado de los mismos, y una interfaz de salida digital o analógica. Los productos de este tipo en el mercado son algo limitados, encontrando actualmente disponibles los modelos TLS3301CL [52] y TLS1401CL [53]. El primero de ellos presenta un conversor A/D integrado y muestra una salida digital de 8 bits por píxel, usando SPI. El segundo tiene una salida directamente en valor analógico, lo que supondría la necesidad de utilización de un ADC o un microprocesador con entradas analógicas. El precio de estos productos se encuentra en torno a los 10€.

Las ventajas de este grupo son, como en el anterior, la flexibilidad total de la que se dispone, teniendo un tamaño pequeño e interfaces fácilmente asumibles por cualquier microprocesador. En este caso el acondicionamiento electrónico parece ser bastante más sencillo que el correspondiente a los sensores CMOS o CCD. Además, se utiliza un número no muy elevado de píxeles, lo cual limita menos la necesidad de recursos y, por tanto, el consumo del sistema.

Sin embargo, el hecho de que sea un array lineal hace que surjan preguntas en cuanto al acondicionamiento óptico, sobre todo la de si merecería la pena perder buena parte de la información del patrón circular del *speckle*. Además, el hecho de que sean menos píxeles hace que el tamaño de cada uno se multiplique aproximadamente por un factor de 10 (comparando los datos del Excel), lo cual supone también cuestiones acerca de si sería suficiente con ese tamaño para detectar correctamente el patrón de variación. No obstante, como se ha comentado anteriormente, en [42] se utiliza exitosamente un componente como estos para realizar la detección del patrón de *speckle*.

Repasando los principios de diseño, las necesidades principales, si el sistema se quiere integrar como un *wearable* de pequeño tamaño, comunicando la cámara con un microprocesador, pasan por que la cámara sea modular, es decir, se pueda adaptar a las necesidades de procesamiento y sea de pequeño tamaño.

Por lo tanto, comparando los diferentes elementos tras las ventajas y las desventajas desarrolladas para cada uno, las alternativas óptimas son dos. En primer lugar, si se quiere continuar utilizando una cámara comercial, la mejor opción es la de la cámara de **Adafruit** [50], ya que puede ser integrada fácilmente en plataformas de desarrollo.

Por otra parte, se podría buscar la simplicidad del sistema empleando *arrays* de fotodetectores como **TLS3301CL** [52] y **TLS1401CL** [53].

No obstante, al haber escogido como procesador la **Raspberry Pi Zero W**, convendrá utilizar la cámara diseñada para dicha plataforma [51].



Figura 28. Raspberry Pi Zero W y su cámara compatible.

4.3. Batería y gestión energética

4.3.1. Consumo del sistema

Para caracterizar el consumo de la Raspberry Pi Zero W en diferentes situaciones, se definen los siguientes estados que puede atravesar el sistema en su ciclo de funcionamiento:

- **Start:** encendido del sistema, dura aproximadamente 10 segundos.
- **Import:** comienzo del programa de monitorización, donde el procesador realiza el trabajo de importar librerías de Python.
- **Procesamiento:** período de máximo rendimiento del dispositivo, con adquisición de imágenes y procesado.
- **Idle esperando BLE** (Bluetooth Low Energy): etapa del sistema en la que este está a la espera de recibir algún comando mediante BLE.
- **Idle WiFi y Bluetooth:** sistema encendido con WiFi y Bluetooth activados.
- **Idle Bluetooth:** sistema encendido con Bluetooth activado.
- **Idle:** sistema encendido con todas las conexiones desactivadas.

Con estas diferentes situaciones, se toman medidas de la corriente consumida por el sistema, alimentando el mismo mediante una batería Li-Po disponible en el laboratorio, y colocando un amperímetro para la medida de corriente. Aunque utilizando un multímetro como amperímetro no permite tener una resolución temporal del consumo a nivel de eventos del microprocesador, sí permite obtener una medida aproximada de la corriente media que consume el equipo en un período de tiempo.

Con esta configuración, se obtienen los resultados que aparecen en la Figura 29.

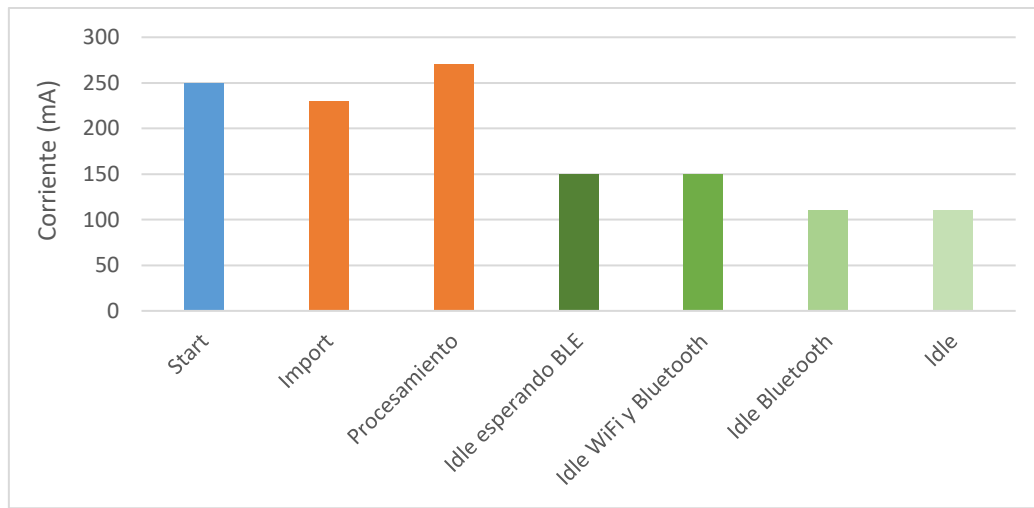


Figura 29. Consumo del sistema en diferentes situaciones.

El consumo pico observado es de 270mA durante la ejecución del programa de procesamiento. Se puede observar que la activación de WiFi supone un aumento en el consumo de 40mA, por lo que, al no emplear esta tecnología por el momento, se mantendrá desactivada en la aplicación final, ya que no se prevé que se vaya a utilizar.

4.3.2. Diseño de batería

Como se ha comentado anteriormente en referencia a la Tabla 1 y la Figura 6, las baterías de litio están generalmente por encima del resto en cuanto a volumen y peso, por lo que, atendiendo a estos factores, serán las más adecuadas para el sistema a diseñar. Además, el voltaje nominal de las mismas, de 3,6V es un valor con el que se puede alimentar a la Raspberry Pi Zero W directamente sin necesidad de convertir a 5V, ya que dispone de un convertidor interno que convierte a 3,3V la tensión para la operación del procesador [54]. Aunque las especificaciones de la tensión de entrada sean 4,75V-5,25V, el sistema funciona correctamente alimentándolo con una batería de litio, ya que estos valores límite vienen dados por las especificaciones de USB [55]. Por tanto, alimentando el sistema con una batería de litio no sería posible conectar ningún dispositivo al puerto USB OTG (*On-The-Go*) disponible en la placa.

El inconveniente de las baterías de litio aparece cuando se manejan de forma incorrecta, ya que presentan un riesgo de explosión ante la sobrecarga y el cortocircuito. Como demostración, se pueden consultar las numerosas noticias en relación a las explosiones de numerosos dispositivos de un modelo de *smartphones* de Samsung, donde debido un fallo de diseño, ambos electrodos de la batería cortocircuitaban [56]. Es por tanto importante tener en cuenta la necesidad de implementar la seguridad ante sobrecargas y cortocircuitos mediante un circuito de protección [15]. De manera redundante, en el circuito de carga de la batería también se suele añadir protección ante sobrecarga.

En cuanto al resto de factores, las baterías de litio presentan en general mejor rendimiento. Aunque sean potencialmente peligrosas ante errores de uso, presentan una toxicidad

relativamente baja frente al resto, además de no necesitar mantenimiento alguno en su ciclo de vida.

Para tomar una decisión entre las baterías Li-Ion y Li-Po, ambas con funcionamiento y rendimiento muy similares, se puede entrar en otro factor de comparación, que es el de la geometría de las celdas. En el caso de las baterías Li-Ion, las celdas suelen tener en su gran mayoría el mismo volumen y geometría, tratándose de cuerpos cilíndricos con una longitud de 65,2mm y un diámetro de 18mm (Figura 30).



Figura 30. Geometría de una celda Li-Ion [57].

Por otra parte, las celdas de Li-Po tienen una geometría rectangular, con un grosor y unas dimensiones que aumentan con la capacidad. Por lo tanto, en función del resultado obtenido respecto al consumo se puede tomar como referencia una operación continua de 7 horas, para lo cual la capacidad necesaria de la batería será:

$$270mA \cdot 7h = 1890mAh \approx 2000mAh,$$

que se redondea a 2000mAh para poder disponer de un valor comercial. Como se puede comprender, 7 horas de funcionamiento continuo se van a dar en casos aislados, y de cara a una primera aproximación del sistema, una batería de la capacidad calculada no supone un problema en cuanto a tamaño y geometría.

En torno a este tamaño, las baterías Li-Po disponibles en el mercado (Figura 31) rondan los 40x60mm, con un grosor de aproximadamente 1mm [58]. Si se compara su geometría con la de la batería Li-Ion, es evidente que, a la hora de agrupar todos los elementos del sistema, una batería “tamaño tarjeta” es más compatible en la integración, resultando un sistema menos aparatoso. No obstante, para ambos casos se tienen baterías que son algo superiores a las dimensiones de la Raspberry Pi (65x30mm), por lo que la batería será el elemento limitante en cuanto al tamaño del sistema.



Figura 31. Batería Li-Po comercial con las dimensiones especificadas.

4.3.3. Gestión energética

Por lo tanto, habiendo decidido el tipo de batería a emplear en el sistema, se debe diseñar ahora su gestión. Debido a la multitud de baterías disponibles con circuitos de protección integrados, se ignorará el diseño de esta parte, de tal forma que la batería dispondrá de un circuito de protección ya dimensionado para dicho producto. Por lo tanto, a continuación, se van a comentar las decisiones de diseño tomadas respecto a la gestión de la carga y la monitorización de la batería, que supondrán una parte importante del hardware del sistema.

En primer lugar, la **carga de la batería** se realizará mediante un integrado dedicado a dicha función. A pesar de disponer de integrados de carga de hasta 1 A, y que la batería escogida [58] tiene la capacidad de soportar dicha corriente de carga, se escoge un cargador programable por una resistencia hasta los 500mA en carga, concretamente el modelo MCP73831 de Microchip [59]. Este controlador de carga está especialmente diseñado para cargar baterías de litio mediante USB, orientado principalmente a dispositivos portátiles dado su pequeño tamaño. Además, la sencillez del circuito a realizar hace que este dispositivo sea perfecto para la aplicación que se le quiere dar en este trabajo (Figura 32).

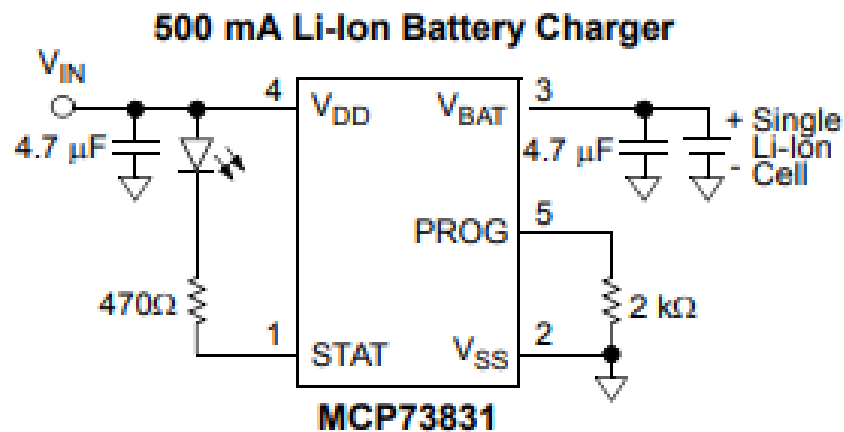


Figura 32. Circuito recomendado para el MCP73831.

La resistencia entre los pines 5 y 2 se utiliza para programar la corriente de carga de la batería, según la gráfica de la Figura 33.

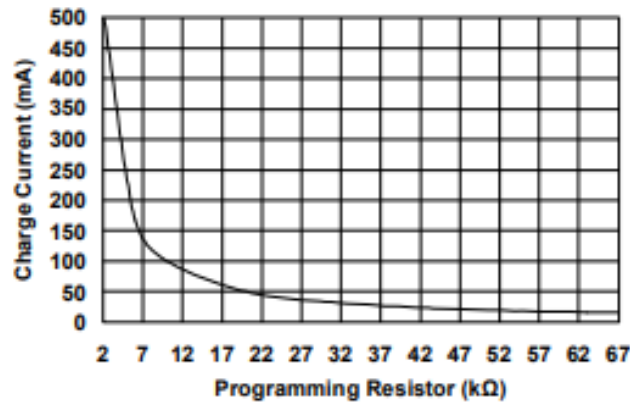


Figura 33. Relación corriente-Rprog.

Por otra parte, con el fin de que el usuario conozca el nivel de la batería en todo momento, se propone implementar un **sistema de monitorización** de ésta. Como se comentó anteriormente en el estado del arte, en el apartado 2.1.2, los contadores de coulomb han evolucionado hasta el punto de no solo detectar la corriente que entra o sale de la batería, si no que existe la posibilidad de monitorizar el voltaje de la batería, y computar su estado de carga (SoC). Los integrados MAX1726X de Maxim [60] combinan un contador de coulomb con el modelo algorítmico propietario ModelGauge m5, utilizando un algoritmo que minimiza los errores producidos por cada técnica.

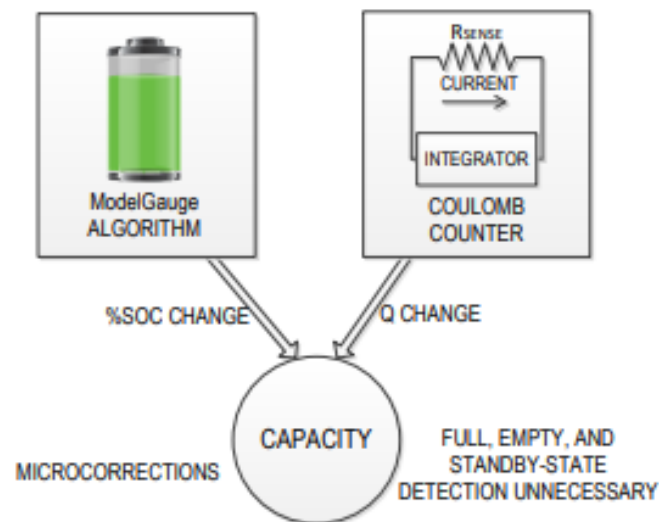


Figura 34. Fusión entre contador de coulomb y el modelo ModelGauge m5.

De esta manera, el integrado MAX17260 tiene capacidades de comunicación por I²C, por lo que la presencia de registros hace que sea capaz de almacenar el valor del nivel de batería y multitud de parámetros que pueden resultar útiles para la gestión energética. El sistema funciona recibiendo cierta información desde el procesador: el voltaje de vaciado, la capacidad nominal de la batería, la presencia o no de un termistor, u otros parámetros de configuración del modelo algorítmico. El integrado automáticamente extrae la información del estado de la batería teniendo en cuenta la edad de la batería, la temperatura y la tasa de descarga que se produce.

El circuito propuesto de la Figura 35 [60] es el que se implementará en la aplicación final, prescindiendo de la resistencia adicional de la parte superior.

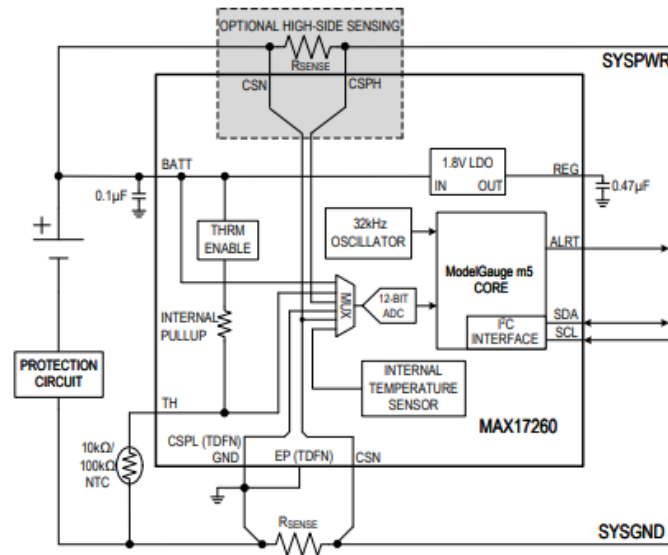


Figura 35. Circuito recomendado para MAX17260.

Sin embargo, la indisponibilidad del integrado MAX17260 en los diferentes canales de venta en el momento de realizar la compra, obligó a escoger una alternativa que no cambiase excesivamente el diseño electrónico que más tarde se presentará. Afortunadamente, el integrado MAX17263 [61] estaba disponible en el momento. Este integrado es prácticamente como el aquí comentado, con dos funcionalidades más:

- Driver de LEDs mediante un botón para visualizar físicamente el nivel de batería.
- Posibilidad de monitorizar varias celdas en serie.

Dada la disponibilidad de este componente, se tomó la decisión de utilizarlo del mismo modo en el que se tenía previsto usar el MAX17260, ya que el conexionado es el mismo, exceptuando los pines de control de los LEDs, que en el integrado inicialmente propuesto son pines sin conexión. Por lo tanto, se utilizó el integrado MAX17263 en su lugar, sin necesidad de realizar modificaciones en el diseño.

Adicionalmente, volviendo a la carga de la batería, se propone un pequeño circuito de *switchover* (Figura 36) que desconecta el sistema de la batería en caso de que se conecte el cargador USB, alimentando el sistema a partir de éste. Para una tensión BAT+ de máximo 4,2 V y USB+ de 5 V, si está conectado el cargador, el voltaje VCC será prácticamente el mismo que en USB+, por lo que el P-MOSFET no conduce y se aísla la batería, que se encontrará en proceso de carga. Sin embargo, cuando se desconecta el cargador y USB+ pasa a ser cero, se aprovecha el paso de la corriente hacia VCC mediante el diodo parásito del transistor, teniendo ahora un valor en VCC prácticamente igual que en la batería. De esta manera, se evita que se produzcan problemas al cargar y descargar la batería al mismo tiempo.

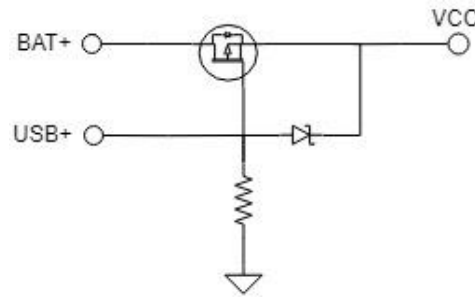


Figura 36. Circuito de switchover.

Por lo tanto, en la Figura 37 se puede consultar un diagrama de bloques del sistema finalmente implementado. El integrado MAX1726X trabaja, como se ha comentado, tanto como contador de coulomb (véase la conexión realizada en la resistencia de sensado) o como medidor de voltaje.

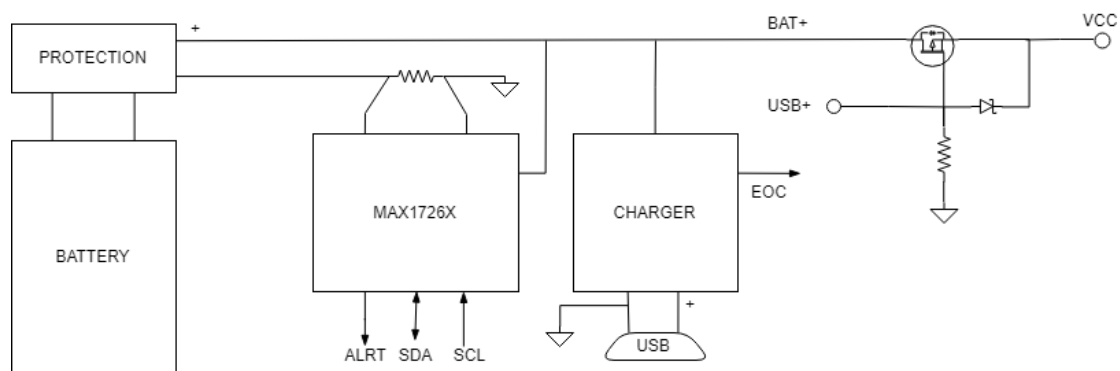


Figura 37. Sistema completo de carga y monitorización de la batería.

4.4. Lógica de funcionamiento

Continuando con el desarrollo del hardware del sistema, se implementan una serie de circuitos adicionales que asegurarán la correcta gestión del apagado y el encendido del mismo, junto a los diferentes circuitos ya comentados de carga y de monitorización de batería.

Parece lógico suponer que, para apagar y encender la Raspberry Pi Zero W, será suficiente disponer de un interruptor que corte y abra la línea de alimentación. Sin embargo, el sistema presenta comunicaciones a nivel interno y externo, y resulta preciso plantear una alternativa, de manera que el procesador cierre todas las comunicaciones antes de apagarse. De esta manera las posibilidades de dañar o reducir el tiempo de vida los componentes asociados se reducen considerablemente. Esto supone, evidentemente, que sea un software el que controle el apagado del sistema.

El problema que presenta la Raspberry Pi Zero W al lanzar un comando de apagado es que entra en un modo “*shutdown*”, donde se consumen 50mA. Este valor comprobado experimentalmente mediante el mismo método que en el apartado 4.3.1 resulta

inaceptable para un sistema alimentado por batería, ya que en unos pocos días se descargaría.

Por lo tanto, para solucionar este problema, se propone un sistema como el planteado en la Figura 38. En este circuito, un pulsador conectado a la señal de reloj de un biestable cambia la salida de éste al producirse un flanco de bajada (pulsación del botón). Esta salida está conectada a la entrada de una puerta OR, que a su vez abre y cierra con su salida el circuito de alimentación del procesador.

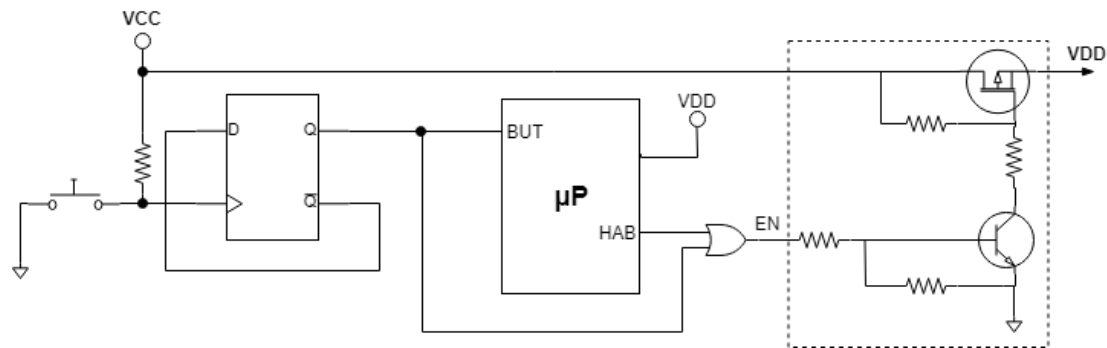


Figura 38. Esquema general de lógica de funcionamiento.

Es en la interacción con el procesador donde este circuito cobra sentido. Cuando el sistema esté encendido, funcionando de manera normal, y se pulse el botón, el procesador detectará este evento ($BUT=0$), y estará comandado por software a cerrar todas sus comunicaciones y terminar los procesos. Mientras esto ocurre, aunque la salida del biestable esté a nivel bajo, la salida $HAB=1$ del procesador permitirá que la puerta OR pueda seguir dando un nivel alto a su salida y el sistema esté alimentado mediante VDD. Cuando el procesador haya terminado de cerrar todos los trabajos críticos, pondrá $HAB=0$, de manera que la salida de la puerta OR pasará a ser un nivel bajo, y por ende el interruptor se abrirá y se cortará la alimentación del procesador.

Cuando se quiera volver a encender el sistema, no habrá más que volver a pulsar el botón, de manera que la puerta OR vuelva a poner su salida a nivel alto y el interruptor se cierre. Se puede leer un resumen del ciclo de funcionamiento a continuación:

Inicio apagado	$BUT = 0, HAB = 0 \rightarrow EN = 0 \rightarrow VDD \neq VCC$
Pulsación ON	$BUT = 1, HAB = 0 \rightarrow EN = 1 \rightarrow VDD = VCC$
Encendido μP	$BUT = 1, HAB = 1 \rightarrow EN = 1 \rightarrow VDD = VCC$
Pulsación OFF	$BUT = 0, HAB = 1 \rightarrow EN = 1 \rightarrow VDD = VCC$
Apagado μP	$BUT = 0, HAB = 0 \rightarrow EN = 0 \rightarrow VDD \neq VCC$

La elección de los elementos como el biestable [62] y la puerta OR [63] se realiza de manera que el consumo sea mínimo, ya que son dispositivos que estarán siempre encendidos, como ocurre con el cargador y el MAX17263.

Este circuito, antes de ser implementado, fue simulado en una placa de prototipado, obteniendo los resultados esperados para los que se diseñó el circuito. No obstante, los componentes que se usaron en las pruebas fueron necesariamente distintos que los finalmente implementados.

Fruto de esta prueba del circuito se decidió colocar un condensador en la entrada de la señal de reloj, de tal manera que los rebotes mecánicos producidos por la pulsación del botón, que provocan una respuesta inesperada del biestable, quedan filtrados y por lo tanto eliminados.

4.5. Diseño de PCB

Todos los circuitos electrónicos que se han comentado en los apartados anteriores estarán implementados en una PCB que, para facilitar el conexionado con la Raspberry Pi Zero W, se ha diseñado con las mismas dimensiones que ésta, de tal manera que se puedan conectar mediante los pines disponibles en dicha placa, a través de un conector de 2x20 pines. En la Figura 39 se puede obtener una mejor visión de esta propuesta de diseño.

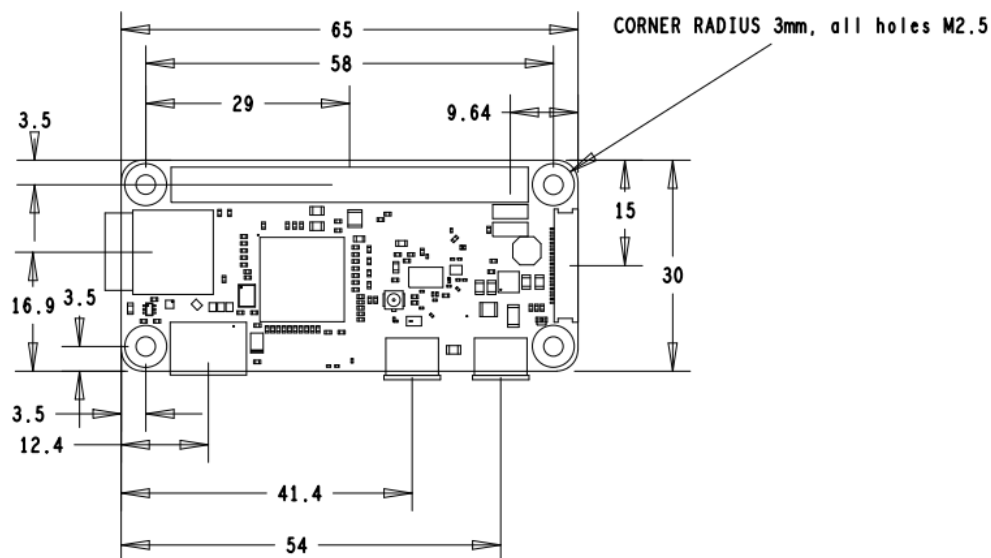


Figura 39. Dimensiones de la Raspberry Pi Zero W.

El esquemático diseñado se puede consultar en el Anexo I. Esquemático PCB, no es más que una integración final de todos los bloques que se han visto: gestión de energía y lógica de funcionamiento, así como la circuitería necesaria para comunicar la Raspberry Pi con el monitor de la batería.

Para el diseño se han tenido en cuenta las instrucciones de los fabricantes de los diferentes chips que componen la placa, de tal manera que se satisfacen todas las condiciones para el buen rendimiento de éstos.

En primer lugar, se puede obtener en la hoja de características del dispositivo MAX17263 [64] una proposición de *layout* para su conexionado en una PCB (Figura 40). Como se

puede comprobar, la resistencia de medida utilizada por el contador de coulomb está conexiónada de manera diferencial, y los condensadores lo suficientemente cerca de los pines para un correcto desacoplo.

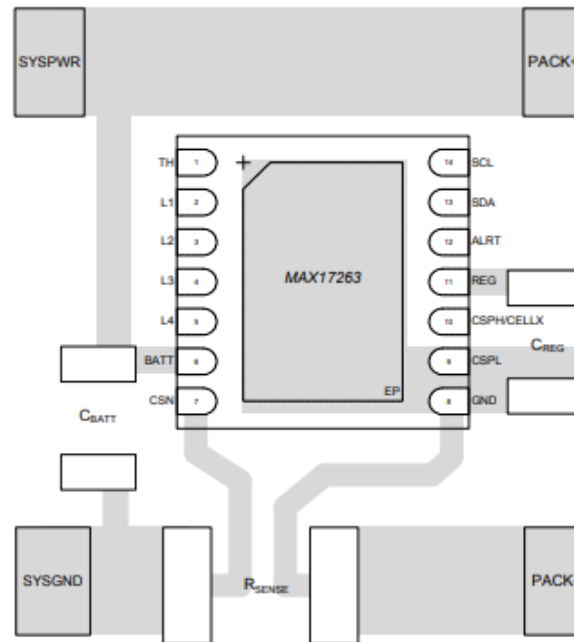


Figura 40. Layout recomendado para MAX17263

Por otra parte, en el *datasheet* del cargador de la batería MCP73831 [59] se propone una disposición de los elementos, utilizando vías entre las capas delantera y trasera de la PCB para disipar el calor producido por el circuito integrado encargado de la carga (Figura 41).

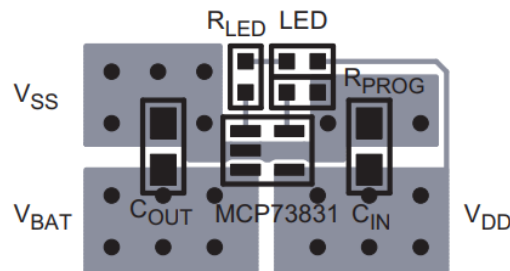


Figura 41. Layout recomendado para MCP73831

El *layout* del circuito finalmente diseñado con la herramienta KiCad [65] se puede observar en la Figura 42, además de incluir un documento en el Anexo II. Layout PCB para su correcta visualización. Adicionalmente, en la Figura 43 se puede comprobar el modelo 3D que se puede generar en KiCad y que sirve para mejorar el proceso de diseño sobre el posicionamiento de componentes.

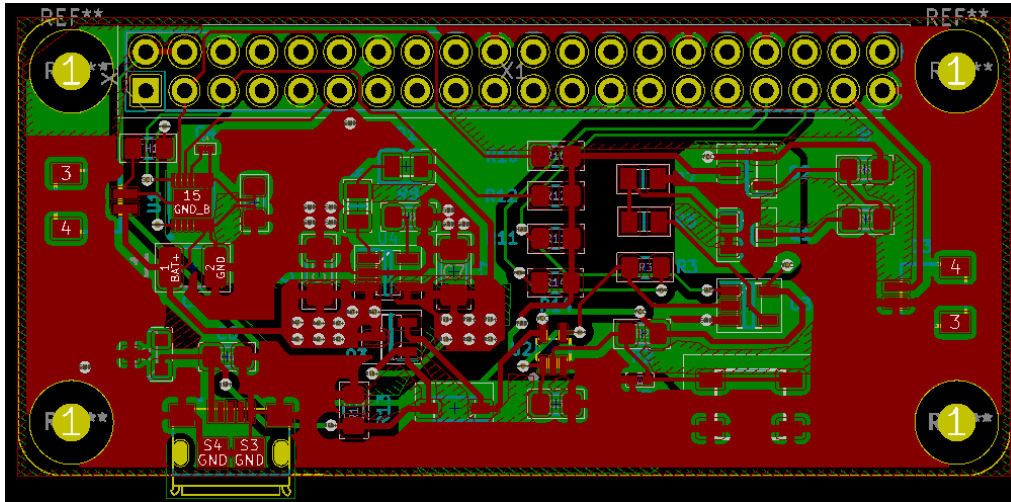


Figura 42. Layout diseñado para la PCB.

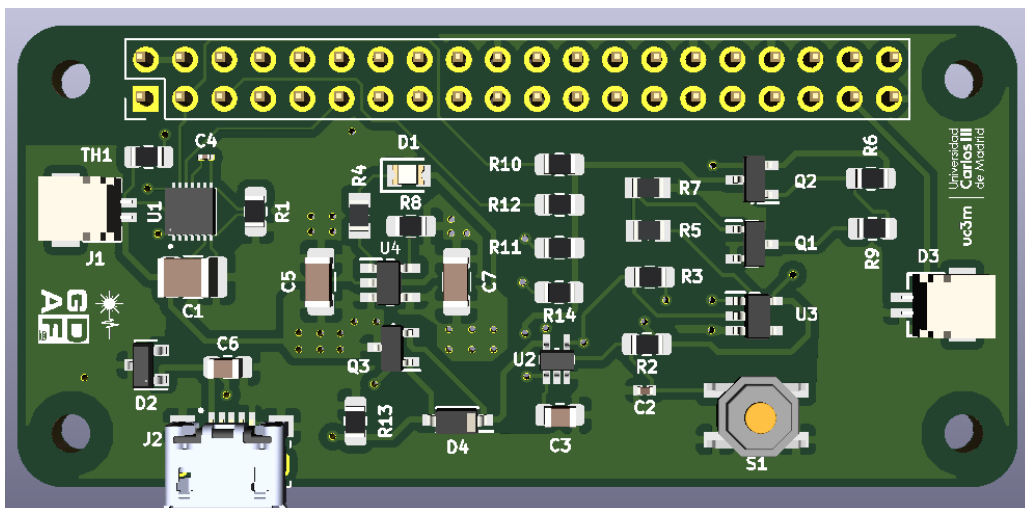


Figura 43. PCB diseñada en visor 3D de KiCad.

Adicionalmente, en la Figura 44 y la Figura 45, el lector se puede hacer una idea del tamaño de la placa que se diseñó, en comparación con una moneda de un euro y con la propia Raspberry Pi Zero W.

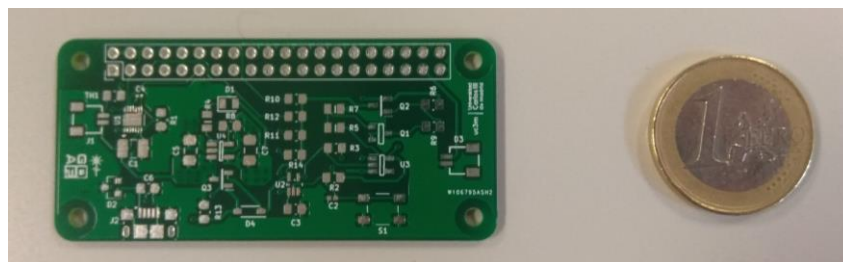


Figura 44. PCB fabricada en comparación con moneda de euro.

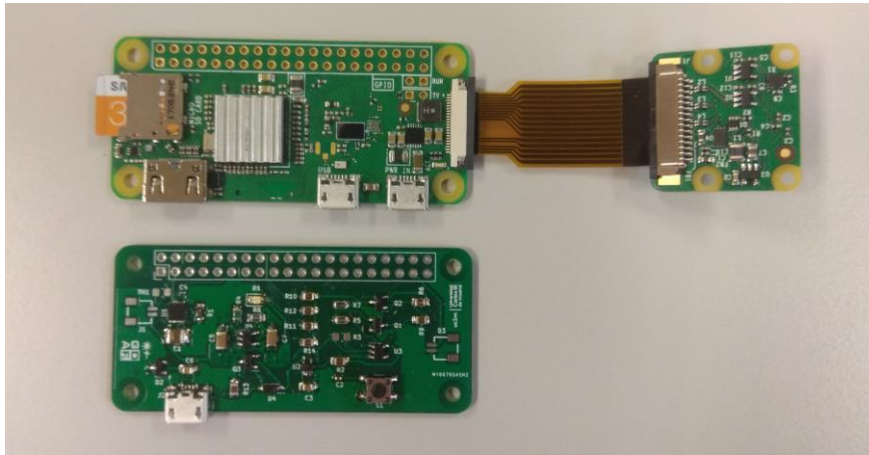


Figura 45. Tamaño de la PCB en comparación con Raspberry Pi Zero W.

Por último, en la siguiente figura se muestra la PCB completamente montada en su versión final, conectada a la Raspberry Pi Zero W, ambas alimentadas mediante la batería.

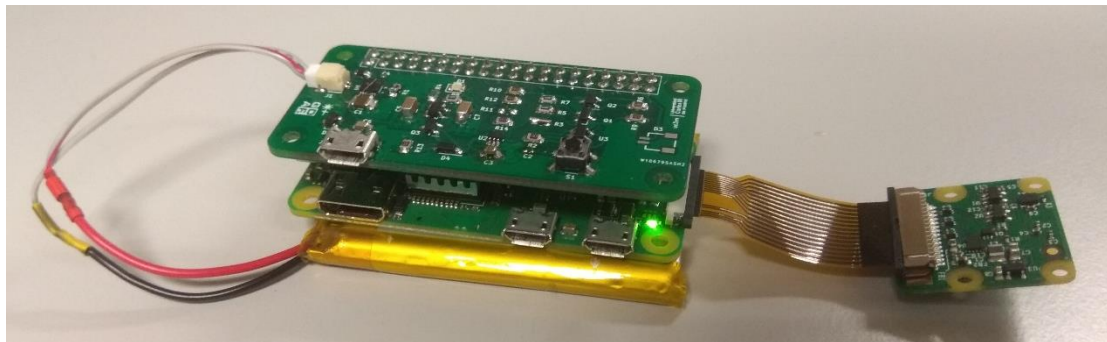


Figura 46. PCB montada conectada a Raspberry Pi.

Para comprobar el correcto funcionamiento de la PCB, se realizan las siguientes pruebas de funcionamiento:

- Funcionamiento de cargador de batería. Se realizan varios ciclos de carga y descarga, realizando medidas de voltaje a lo largo de los mismos. Se comprueba que el LED indicador muestra el final de carga correctamente.
- Funcionamiento de monitor de batería. Primero se comprobará el funcionamiento del bus I²C, para después realizar algunas medidas del nivel de batería a lo largo de los ciclos y extraer conclusiones. Las pruebas y conclusiones se detallan en el capítulo 0 (Pruebas de *hardware*).
- Funcionamiento de la lógica. Se monitoriza el botón de apagado y encendido mediante interrupciones, pudiendo realizar acciones si se pulsa el botón. Las pruebas y conclusiones se detallan en el capítulo 0 (Pruebas de *hardware*).

4.6. Diseño 3D de carcasa para integración

La última parte del diseño del hardware consiste en la integración de todos los elementos dentro de un espacio cerrado o carcasa: las PCBs, la cámara, el láser y la fibra deben estar correctamente sujetos para que el sistema pueda operar de manera óptima. La disposición de los elementos en el programa FreeCAD [66], colocados de acuerdo a sus medidas, se puede observar en la Figura 47.

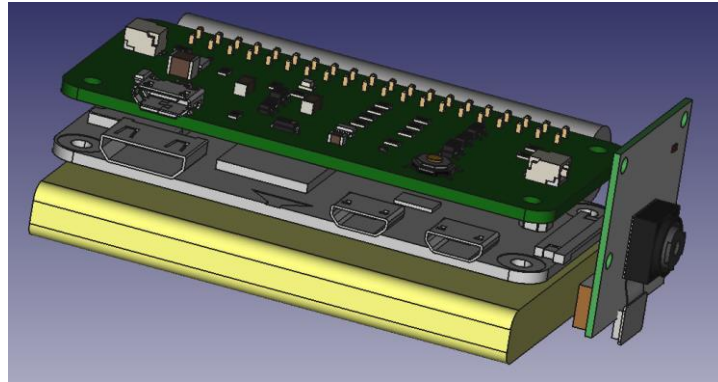


Figura 47. Disposición de los elementos en FreeCAD.

El modelo de la PCB diseñada se ha exportado de la propia plataforma KiCad, tanto la batería como el láser (cilindro en la parte trasera) se han diseñado de acuerdo a sus medidas mecánicas experimentales. Por otro lado, tanto la Raspberry Pi Zero W como la cámara se han obtenido de la plataforma Thingiverse para poder tener una referencia fiable en FreeCAD [67].

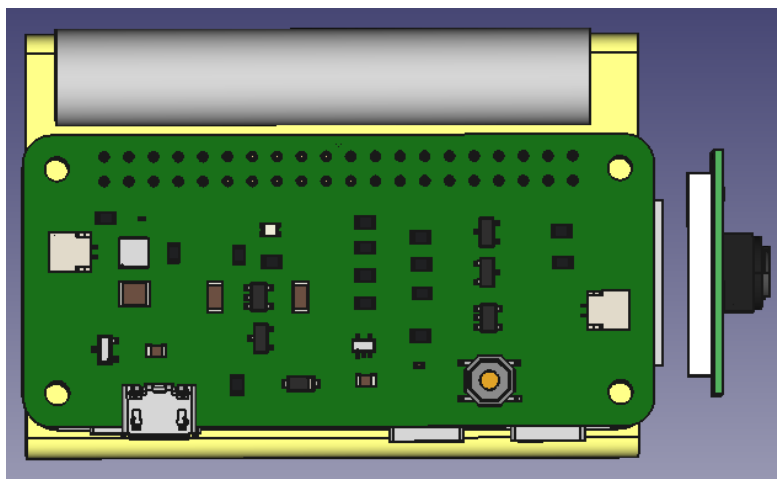


Figura 48. Vista en planta de la agrupación de los elementos en FreeCAD.

Mediante la vista en planta de la Figura 48 se puede ver cómo se pretende aprovechar el exceso de área que cubre la batería, mediante la colocación del láser, el cual está ya acoplado a la fibra. No obstante, se deberá asegurar la sujeción y ausencia de holgura o movimientos de todos y cada uno de los elementos para no distorsionar la señal de *speckle*.

Como diseño preliminar de la parte que se encargará de la correcta disposición tanto de la electrónica como de la óptica, se propone lo siguiente, realizado con el software FreeCAD:

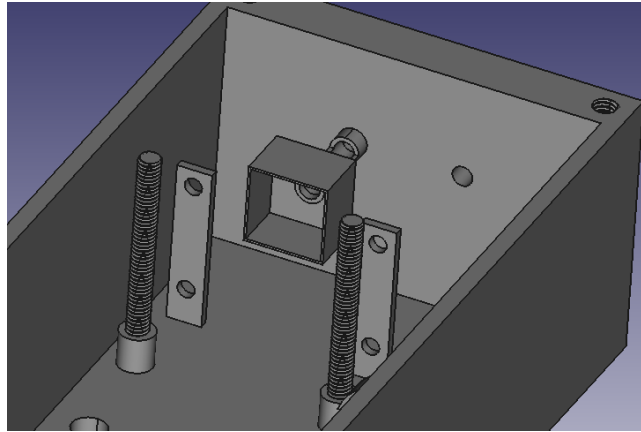


Figura 49. Diseño inicial 3D. Vista cámara.

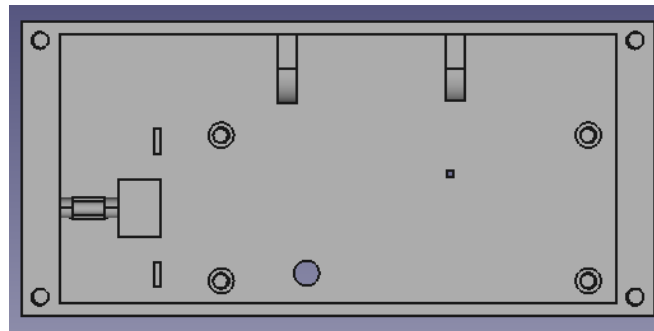


Figura 50. Diseño inicial 3D. Vista planta.

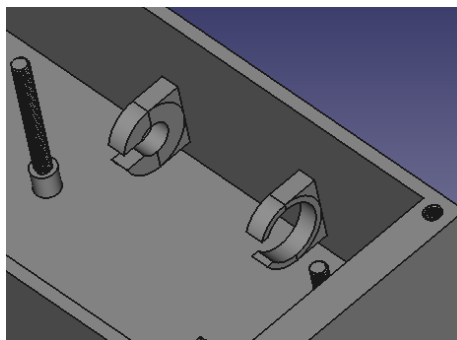


Figura 51. Diseño inicial 3D. Vista láser.

Como se puede comprobar en la Figura 49, se ha propuesto un mecanismo de acoplo a la cámara, de manera que se asegure que la fibra esté totalmente enfrentada al sensor, además de asegurar una sujeción perfecta de la cámara. Por otra parte, se puede observar también como se ha provisto a la estructura de cilindros roscados que se utilizarán para sujetar las PCBs mediante una tuerca de métrica 2,5. Además, en la Figura 51 se comprueba cómo se han colocado dos estructuras que actuarán como sujeción del láser de LasIRvis comentado previamente (ver Figura 24), de acuerdo con sus medidas— presenta dos diámetros distintos.

En cuanto a la parte que sujetará la batería, ésta actuará como el elemento que cierra el sistema, de tal manera que se dispondrán de los huecos para introducir tornillos de montaje, tanto para unir las dos partes de la carcasa, como para sujetar la batería. Se puede obtener una idea más clara del diseño en la Figura 52

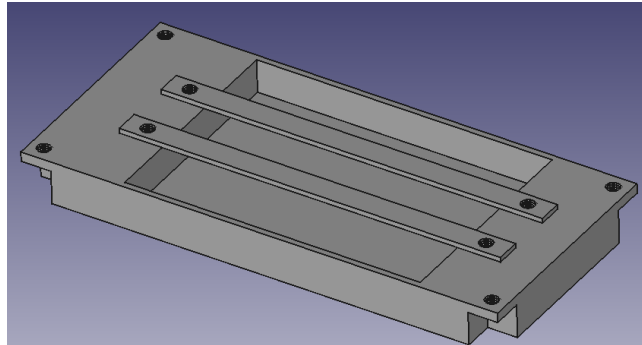


Figura 52. Diseño inicial 3D. Sujeción batería.

Como se puede observar, se dispondrá de un hueco donde se colocará la batería, además de dos elementos para sujetarla. Es importante asegurar que la batería no presente ningún tipo de movimiento, ya que es el elemento que más pesa del sistema, y por ende el que más fácilmente podría distorsionar la señal de *speckle*.

Finalmente, en la se puede observar cómo quedaría finalmente la carcasa completa en su diseño preliminar, incluyendo las medidas más características. Se han realizado algunos huecos para diferentes elementos: fibra, luces, conector USB y botón.

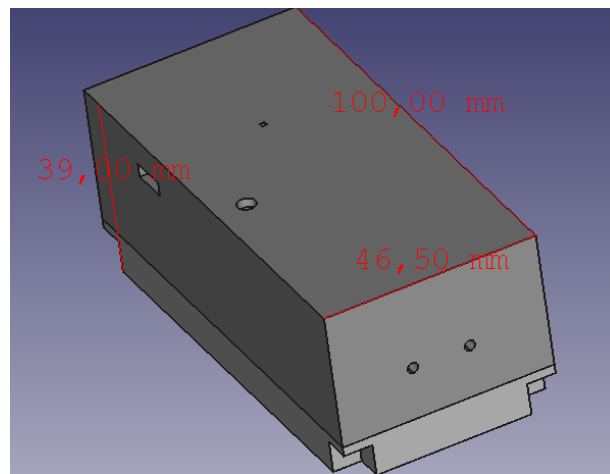


Figura 53. Diseño inicial 3D. Carcasa completa.

Capítulo 5

Diseño del software

El diseño y el desarrollo del software para el presente sistema presenta dos partes bien diferenciadas. Por un lado, se ha llevado a cabo el desarrollo de un sistema de procesamiento de señal en tiempo real, con el fin de que se pueda ser implementar en la Raspberry Pi Zero W. Por el otro lado, se ha realizado el diseño de los elementos de interfaz de usuario, compuesto principalmente por un sistema de comunicación BLE implementando un periférico o servidor en el sistema portátil y un cliente en una aplicación en Android. Esta última funcionalidad incluirá modificaciones dentro del software de procesamiento de datos con el fin de establecer una comunicación correcta en tiempo real. Por lo tanto, se comentará inicialmente el funcionamiento de los procesos de adquisición y procesamiento de las imágenes, pasando posteriormente a comentar tanto la aplicación servidor y las modificaciones o extensiones introducidas en los programas anteriores, como la aplicación cliente en Android.

5.1. Software de procesamiento de datos

En este apartado se han ido tomando aproximaciones graduales, consiguiendo replicar lo realizado hasta el momento en MATLAB con la cámara PS3 Eye, de una manera óptima en un sistema con mayores limitaciones que un PC.

La primera decisión al respecto es la de escoger un lenguaje en el que la migración del código de MATLAB pueda llevarse a cabo de manera sencilla. Las principales opciones que ofrece el sistema operativo de la Raspberry Pi, Raspbian (basado en Debian), son C++ y Python. Es evidente que un lenguaje compilado como C++ ofrece una fluidez de ejecución mejor que la de un lenguaje interpretado como Python. Sin embargo, Python es un lenguaje muy utilizado por la comunidad científica [68], con librerías como Numpy o SciPy que permiten desarrollar código de manera similar a MATLAB, facilitando así el procesamiento de datos, principal propiedad del software aquí presentado.

Tras esto, es importante tener en cuenta la principal diferencia respecto al código de MATLAB utilizado en las pruebas. De cara a la implementación final, se buscará que el software ofrezca un procesamiento en tiempo real del ritmo cardíaco. Con anterioridad se habían realizado medidas *a posteriori*, grabando vídeos con una cámara y procesando las imágenes posteriormente en MATLAB. Aquí se buscará paralelizar los procesos de adquisición y procesamiento de datos para llegar al objetivo planteado. Por este motivo, se explicarán a continuación ambos programas por separado, y posteriormente, en el epígrafe 5.2.1 se explicará el procedimiento realizado para sincronizar y comunicar estos procesos.

5.1.1. Programa de adquisición y pre-procesado

Este primer proceso tiene dos partes diferenciadas: una primera, en la que se inicializa la configuración de la cámara, estableciendo los parámetros de ésta para la correcta visualización del patrón de *speckle*, y una segunda parte en la que se graban las imágenes a la frecuencia deseada, se comparan y se obtiene la señal unidimensional resultante que se enviará al programa de procesamiento.

A modo de ilustración, se puede consultar la Figura 54, donde se presenta un diagrama de flujo del presente proceso.

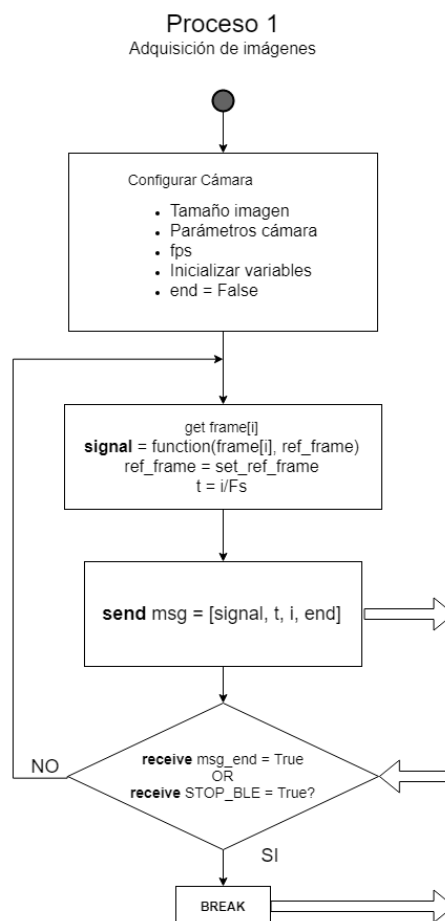


Figura 54. Diagrama de flujo del proceso de adquisición.

Como se puede observar, al principio del proceso se inicializa el sistema, para después entrar en un bucle donde, cada cierto tiempo, el procesador capturará y pre-procesará una imagen con el fin de obtener un valor de una sola dimensión y poder enviar este valor al sistema de procesamiento (comunicación simbolizada por las flechas gruesas). Cuando se haya sobrepasado el tiempo deseado o el usuario haya solicitado la parada del procesamiento mediante la aplicación, el proceso terminará, haciéndoselo saber antes al programa de procesamiento de datos (*end = True*).

La configuración inicial ajusta todos los parámetros de la cámara, que condicionarán las imágenes que posteriormente se procesarán. El principal parámetro de la cámara es el balance automático de blancos (*awb_gains*) [69], utilizado principalmente para añadir una seguridad adicional de que la cámara va a captar la imagen del *speckle*, ya que según el valor designado, la cámara se adaptará automáticamente a la luz ambiente, filtrándola correctamente y captando una señal de *speckle* con un contraste determinado. Esta propiedad de la cámara simplifica de manera considerable su puesta a punto.

Por otra parte, según el procesamiento que se utilice, se utilizarán un tamaño de imagen determinado, de manera que el procesador no se sature. Esto se debe a que el tiempo de duración de cada iteración bucle vendrá determinado por la frecuencia de muestreo, F_s , de tal manera que la duración máxima de cada iteración será:

$$t_{max} = 1/F_s \quad (9)$$

Esto supone una limitación a la hora de realizar las operaciones correspondientes para obtener la señal a procesar posteriormente, dado que, para una frecuencia de muestreo de 10 fps (*frames per second*), el tiempo máximo para realizar las operaciones es de 0,1 segundos. Por lo tanto, aunque en MATLAB se hubiera utilizado la función correlación entre una serie de imágenes anteriores y la actual, en la aplicación final en tiempo real aparecen limitaciones como la presente, que dificultan realizar operaciones de gran tamaño (como es la correlación entre imágenes) en un tiempo ajustado.

Las diferentes alternativas vienen simbolizadas en el diagrama de la Figura 54 como *function* y *set_ref_frame* que, respectivamente, indican la operación realizada para obtener la señal y la manera de obtener una imagen de referencia para realizar la comparación después. Las cuatro alternativas propuestas son las siguientes:

1. Diferencia entre imágenes. Esta alternativa resulta la manera más sencilla de representar la diferencia entre dos imágenes, y consiste en restar la imagen actual y la anterior en su componente roja, realizando después una media de la “imagen resta” resultante, en la línea de lo implementado en [40]. La operación *function* realizada es la siguiente:

$$s = \frac{\sum_m \sum_n (A_{m,n} - B_{m,n})}{m \times n} \quad (10)$$

donde s es la señal, m y n representan la resolución de la imagen, A es la imagen actual y B la imagen directamente anterior (en sus componentes rojas). Respecto a *set_ref_frame*, en este caso la imagen utilizada como referencia es la imagen anterior, por lo que cada vez que se realiza la operación, se actualiza *ref_frame* con el valor de la imagen actual, que será utilizado en la siguiente iteración.

En esta alternativa, las limitaciones respecto al tiempo de adquisición no suponen un problema debido a la sencillez de la operación, y se puede emplear la cámara con una resolución de aproximadamente 300x300 píxeles.

- 2. Intensidad de imagen.** Ante la existencia de sensores de ritmo cardíaco basados en PPG (epígrafe 2.2), interesa aplicar la metodología utilizada sobre dicha tecnología en Nguyen et al. en [70] para comprobar una posible relación de la tecnología actual y la PPG. En este caso, la señal es la intensidad media del rojo en cada imagen, prescindiendo por tanto de la imagen de referencia. La operación *function* realizada es la siguiente:

$$s = \frac{\sum_m \sum_n A_{m,n}}{m \times n} \quad (11)$$

donde s es la señal, m y n representan la resolución de la imagen y A es la imagen actual (en su componente roja).

Como ocurre en la alternativa anterior, la operación a realizar es sencilla, por lo que el tiempo máximo de procesamiento no será limitante.

- 3. Correlación.** Como en el software de MATLAB, en este caso se buscará el valor de la correlación entre la imagen actual y una combinación de las anteriores, que será la imagen de referencia. La operación *function* utilizada es la función *correlate2d* de SciPy [71], que realiza la correlación cruzada entre dos matrices. La primera matriz es la imagen actual, y la segunda matriz es la resultante de *get_ref_frame*; se trata de una media móvil con pesos como la siguiente:

$$ref_frame = \frac{3}{6} A_{i-1} + \frac{2}{6} A_{i-2} + \frac{1}{6} A_{i-3} \quad (12)$$

donde A es una variable donde se guardan las últimas tres imágenes. Como se puede observar, se les da mayor peso a las imágenes más cercanas en el tiempo.

El hecho de que se realice una operación de correlación entre matrices y se tenga que recalculan la imagen de referencia en cada iteración, supone un problema a la hora de entrar dentro de las especificaciones temporales dadas por la frecuencia de muestreo. Por este motivo, la resolución de 300x300 píxeles de los apartados anteriores impide realizar las operaciones en el tiempo establecido, por lo que, para evitar problemas con la integridad de la señal y de la propia cámara, se elige una resolución de 100x100 píxeles.

4. Diferencia de fase (*Phase-Shift*). Esta operación resulta popular en el ámbito de la interferometría por patrón de *speckle*, siendo un método al que se ha recurrido desde hace bastantes años, como se ha comentado anteriormente en el epígrafe 2.3.2. Además, ha sido empleado para la misma aplicación que el presente trabajo [42].

La función *Phase-Shift* establece el cambio de fase del patrón de *speckle* en las tres últimas imágenes adquiridas, píxel a píxel, y realizando una suma para obtener la señal unidimensional:

$$\theta_i(x, y) = \arctan \left[\sqrt{3} \frac{p_i(x, y) - p_{i-1}(x, y)}{2p_{i-2}(x, y) - p_{i-1}(x, y) - p_i(x, y)} \right] \quad (13)$$

$$s = \sum_m \sum_n \theta_i(m, n) \quad (14)$$

Donde s es la señal a obtener, $\theta(x, y)$ el cambio de fase de un píxel, $p_n(x, y)$ el valor de rojo de un píxel y m, n representan la resolución de las imágenes.

Aquí por lo tanto tampoco hay un *ref_frame* como tal, pero se realiza una comparación entre la imagen actual y las últimas dos imágenes, por los que los valores de dichas imágenes han de actualizarse en cada iteración. Además, como ocurre en el apartado anterior, el peso de la operación es importante, lo que supone un ahorro de resolución a la hora de adquirir imágenes.

5.1.2. Programa de procesado

El segundo proceso, corriendo en paralelo con el de adquisición, es aquel en el cual se aplican los filtros y operaciones necesarias a la señal obtenida anteriormente para determinar el valor del ritmo cardíaco. Para no saturar el sistema, aunque se recopilarán los datos pre-procesados del programa anterior en tiempo real, éstos serán procesados solamente cada cierto tiempo. El diagrama de flujo de este sistema se muestra en la Figura 55.

Cuando el programa comienza, recopilará los datos enviados por el proceso de adquisición, almacenándolos en un arreglo de valores con un tamaño n prefijado. El valor de esta variable es el que define el tamaño de las señales que se van a procesar. Con una frecuencia de muestreo de 10 fps, si se quieren procesar muestras de 20 segundos, el valor de n deberá ser de 200. Por tanto, si el número de valores es menor que n , estos serán almacenados uno detrás de otro en la variable correspondiente, pero cuando alcanzan su valor máximo ($i > n$), será necesario implementar un mecanismo FIFO (*First-In First-Out*), de tal manera que pueda entrar el nuevo valor por el final del *array*, saliendo del mismo el valor que había entrado en primer lugar. De esta manera, se mantiene una ventana de tiempo real de los últimos 20 segundos (en este caso). Se puede comprobar este mecanismo de manera gráfica en la Figura 56.

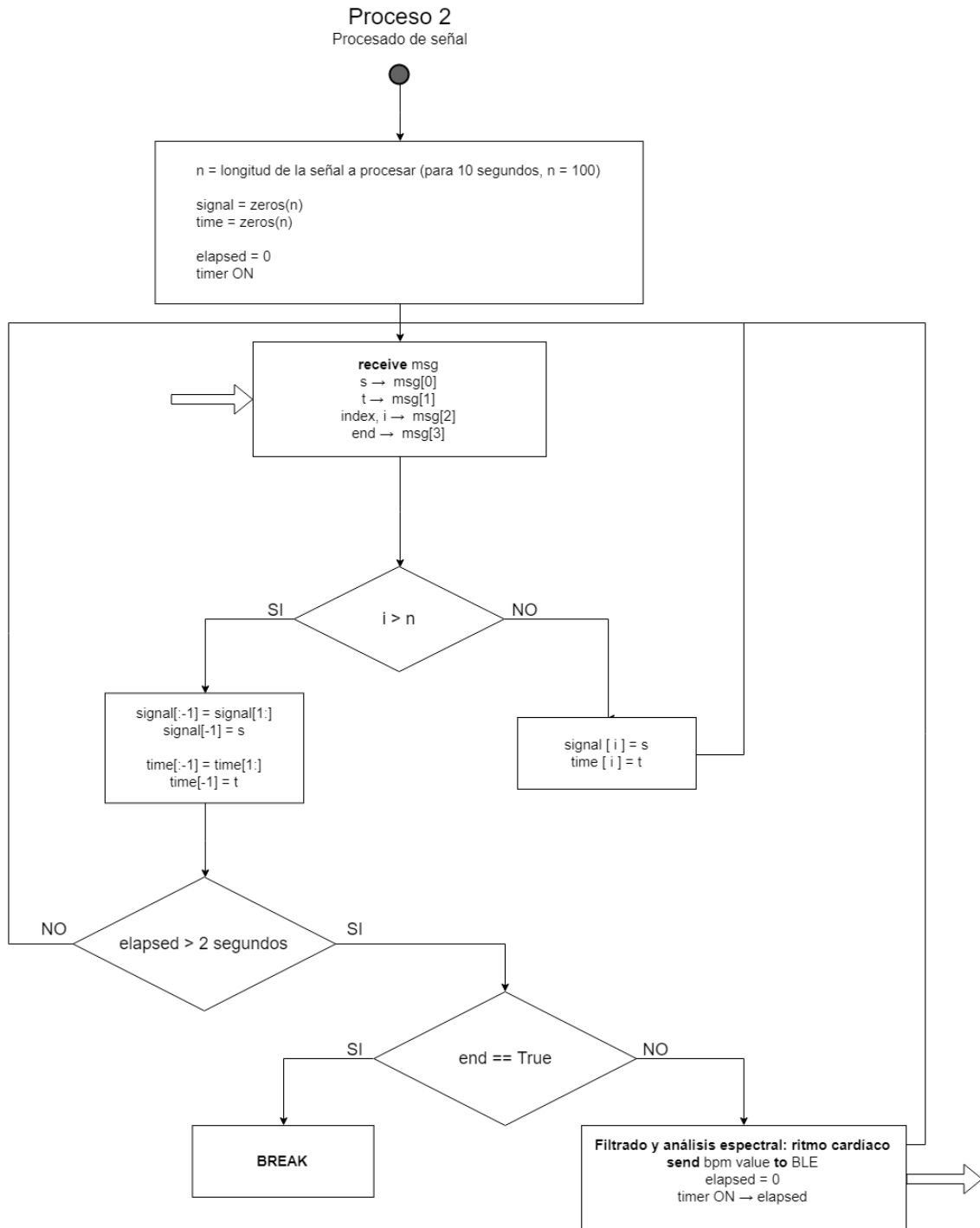


Figura 55. Diagrama de flujo del procesado de señal.

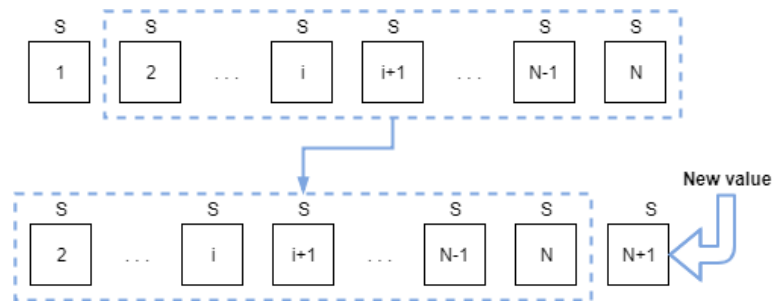


Figura 56. Mecanismo FIFO.

Tras el almacenamiento de la señal actual, el sistema solamente aplicará los filtros y mecanismos de procesamiento cuando se cumplan tres condiciones:

- 1) se debe haber pasado la barrera de los n valores, es decir, la variable correspondiente deberá tener almacenados n valores;
- 2) el tiempo transcurrido entre dos procesamientos debe ser mayor o igual a 2 segundos;
- 3) la variable *end* recibida deberá tener un valor de *False*.

Tras adquirir una ventana de datos (n valores), a la hora de filtrar y extraer el valor de ritmo cardíaco, se han propuesto varias alternativas. Aunque de cara a la aplicación final solo se empleará una de ellas (como en el proceso de adquisición), resulta interesante comentar los diferentes tipos de filtrado y análisis espectral que se han considerado. Por una parte, en cuanto al filtrado, las alternativas son tres: filtro Butterworth, filtro FIR y filtro *Savitzky-Golay*. En el caso del análisis espectral se tienen también tres alternativas: espectrograma, análisis de densidad de potencia espectral (PSD, *Power Spectral Density*) de *Welch* y la función de búsqueda de máximos locales (picos), llamada también *Findpeaks*.

Antes de comentar los diferentes tipos de filtrado y análisis espectral, es importante destacar que, antes del filtrado de la señal, se somete a la misma a la función *detrend*, que elimina cualquier tendencia lineal presente en la ventana de datos, estabilizándola sobre el eje horizontal.

Filtros utilizados

1. **Butterworth.** Filtro de respuesta de impulso infinita (IIR, *Infinite Impulse Response*), que se caracteriza por ofrecer una función de transferencia sin ningún tipo de rizado, obteniendo una señal suavizada a la salida [72].
2. **FIR.** Del inglés *Finite Impulse Response*, este filtro digital tiene varias propiedades que resultan de interés para la presente aplicación: no requieren de realimentación, lo que los hace estables por definición; además, se pueden diseñar para que no se produzcan desfases. Sin embargo, necesitan mayor capacidad de computación que los filtros IIR como el Butterworth, uno de los motivos por los que conviene realizar la

comparación de rendimiento entre los diferentes filtros, de acuerdo con el consumo de recursos del procesador [73].

3. **Savitzky-Golay.** Este filtro se utiliza con frecuencia para suavizar señales por convolución. Como ventaja destaca el hecho de que mantiene los máximos y mínimos locales, es decir, apenas modifica la señal de entrada. Sin embargo, para una señal tan ruidosa como la que se desea filtrar, la aplicación de este filtro puede dar lugar a errores de medida [74].

Métodos de análisis espectral utilizados

1. **Espectograma.** Este método de análisis frecuencial en el tiempo se aplica principalmente en el campo del audio y el reconocimiento del habla. Se trata del resultado de la aplicación de la Transformada de Fourier de Tiempo Reducido (STFT) a ventanas de tiempo con un pequeño solapamiento, obteniendo finalmente un valor en frecuencia para cada ventana de tiempo [75].
2. **PSD Welch.** Este método fue implementado por Peter D. Welch en 1967 [76], y aplica la Transformada de Fourier Rápida (FFT), junto a un algoritmo para extraer periodogramas y obtener la potencia espectral de una señal. Su uso es muy extendido en el ámbito del análisis frecuencial en el tiempo, por lo que resulta preciso incluir esta alternativa en las mediciones.
3. **Findpeaks.** Este método, cuya traducción al español sería “busca picos” se trata de, como su nombre indica, un algoritmo de búsqueda de picos en una señal. Este algoritmo, disponible en la librería SciPy de Python [77], recibe como entrada un array de datos, que en este caso sería la señal a procesar, y devuelve otro array que contiene las posiciones de la señal donde se han encontrado picos. Este método resulta interesante de comprobar, aunque para señales con mucho ruido es fácil que resulte en falsos positivos. Se trata de un método muy empleado para procesar señales del tipo ECG, donde la relación señal-ruido es muy elevada.

5.1.3. Comparativa de alternativas de procesado

A pesar de poder comparar teóricamente los métodos de procesado, es preciso no descartar ninguna alternativa y realizar pruebas en la práctica para poder tomar una decisión final respecto a qué metodología de procesado utilizar en la aplicación final.

Para este fin, se realizarán pruebas con un zumbador electrónico, que se modulará mediante un generador de funciones con la frecuencia deseada y se mantendrá en contacto con la fibra. Las vibraciones mecánicas del zumbador afectarán al patrón de *speckle* de manera similar a como lo hace el pulso cardíaco. Con el fin de desarrollar un sistema lo

más robusto posible, para cada configuración software se realizarán medidas a frecuencias y amplitudes de excitación distintas, teniendo como objetivo la respuesta correcta del sistema a amplitudes bajas.

Es importante destacar que el esfuerzo de realizar estas pruebas y analizar posteriormente las medidas obtenidas lo ha puesto mayoritariamente Ignacio Sánchez, autor principal de [43]. Por lo tanto, en este apartado se dará simplemente una visión general de las pruebas realizadas y sus resultados.

Las pruebas de *software* se configuran de la siguiente manera:

- Se realizarán medidas con cada una de las variantes de adquisición (diferencia, intensidad, correlación y *phase shift*) a diferentes frecuencias y amplitudes, con el objeto de simular lo más fielmente posible la acción del pulso cardíaco.
- Para cada una de estas variantes, en el programa de procesado de datos se llevarán a cabo todos los filtrados y análisis comentados anteriormente, obteniendo 9 valores de ritmo cardíaco correspondientes al total de combinaciones posibles.
- Tras realizar las medidas consideradas, se analizará el error de los valores de ritmo cardíaco obtenidos con respecto a la frecuencia marcada por el generador de funciones.
- Se tomará una decisión de cara a la implementación del *software* en la aplicación final.

Para poder obtener una visión resumida de los resultados obtenidos, se han calculado, para cada método de adquisición, la media del error (en porcentaje) de la mejor combinación filtrado-análisis espectral de la medida. Todo esto se ha realizado para cada voltaje de excitación del zumbador. En la Tabla 4 se pueden consultar los resultados obtenidos.

Tabla 4. Resultados de las pruebas de procesado.

Señal/Excitación	3,5 V	2,5 V	1,25 V
Diferencia	0,66 %	0,66 %	3,96 %
Intensidad	0,66 %	0,66 %	0,55 %
Correlación	12,16 %	-	-
Phase-Shift	1,96 %	3,56 %	2,52%

Como se puede comprobar, los mejores resultados se obtienen comparando la intensidad de las imágenes, obteniendo buenos valores incluso para excitaciones de pequeña amplitud. Sin embargo, el método de diferencia se acerca bastante al anterior, teniendo en ocasiones valores idénticos para algunas configuraciones de filtro-análisis espectral implementadas. Por detrás de ambos, se encuentra el método de cambio de fase o *phase shift*, para el cual se obtienen valores correctos, pero relativamente peores que los dos anteriores.

Por otra parte, es importante destacar que el método de correlación empleado en el sistema anterior resulta bastante errático en este caso. Esto es debido al carácter del procesado que se quiere realizar aquí, en tiempo real. La velocidad de pre-procesado estará limitada por la frecuencia de muestreo de la cámara; además, la operación de correlación entre matrices supone un consumo de recursos importante. Esto llevará a disminuir la resolución de las imágenes capturadas con el fin de evitar problemas en el procesado y poder realizar las operaciones en el tiempo disponible. Al disminuir tanto la resolución de la grabación, el *speckle* se hará menos visible, llegando a un punto en el que es totalmente imperceptible para vibraciones de baja amplitud. Por lo tanto, al obtener valores fuera de lugar se ha decidido no incluir en esta comparativa las medidas obtenidas con el método de correlación para voltajes de 2,5 y 1,25 V.

5.2. Software de comunicación

Como se ha comentado anteriormente en el epígrafe 2.1.2, el *wearable* que se diseña en el presente trabajo se comunica con el exterior mediante un sistema de comunicación inalámbrica que, en este caso es Bluetooth en su versión de bajo consumo, *Bluetooth Low Energy* (BLE). De esta manera, se implementará un periférico sobre Python en la Raspberry Pi Zero W, que se comunicará con un cliente implementado en una aplicación de Android.

No obstante, dado que el presente trabajo se realiza a modo de demostración del sistema completo, esta aplicación estará limitada en cuanto a funcionalidades de cara a asegurar un funcionamiento correcto de la misma, abriendo la puerta a futuras ampliaciones. El usuario será capaz de comenzar y parar la monitorización del ritmo cardíaco, además de poder visualizar por un lado el valor de pulso, y por otro el nivel de la batería.

Sin embargo, para poder sincronizar todos los programas implementados con Python, se debe desarrollar un sistema de comunicación entre los mismos, que es por donde se comenzará a continuación, pasando después a explicar la comunicación implementada con Bluetooth.

5.2.1. Comunicación entre procesos

Para poder paralelizar varios procesos en un sistema basado en Linux, Python pone a disposición la librería *multiprocessing*, ofreciendo diferentes herramientas para lograr cierto nivel de concurrencia con Python [78].

Cada programa será un objeto de la clase *Process*, de manera que se podrán ejecutar varias funciones (correspondientes a cada programa) por separado. Estos procesos se comunicarán mediante elementos de tipo *Queue* [79], que se tratan de tuberías (o *pipes*) que comunican dos procesos, mediante un sistema de semáforos implementado internamente por la librería. Normalmente, un proceso escuchará y el otro pondrá datos dentro de la tubería, de tal manera que se produzca una sincronización correcta cuando un proceso ponga datos y el otro reciba los mismos, realizando las operaciones que sean necesarias.

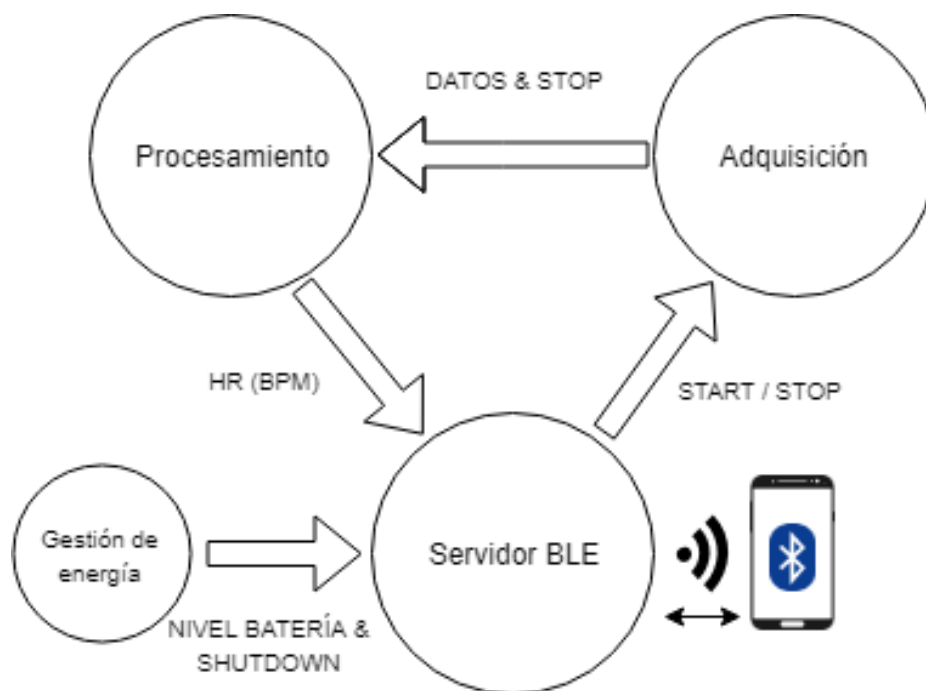


Figura 57. Comunicación de procesos en Raspberry Pi Zero W.

En la Figura 57 se puede tener una visión clara de las comunicaciones implementadas a través de este mecanismo de tuberías, así como de los procesos involucrados. El papel de cada proceso en el funcionamiento del sistema es el siguiente:

- **Adquisición:** este proceso se encargará de la adquisición de la información del patrón de *speckle* presente en la cámara y su pre-procesado para la posterior conversión en una señal unidimensional. Se trata del programa desarrollado en el epígrafe 5.1.1. Se encargará de comunicarse con el Servidor BLE para recibir comandos de inicio y parada de medida que envíe el usuario.
- **Procesamiento:** en este, caso, tal y como se ha desarrollado en el epígrafe 5.1.2, este proceso recibe la señal unidimensional del proceso anterior, y aplica los filtros y herramientas de análisis pertinentes para poder extraer el valor de

frecuencia cardíaca, enviando posteriormente dicha información al usuario a través del Servidor BLE.

- **Servidor BLE:** este programa se trata del canal mediante el cual la información interna puede transmitirse al exterior, por lo que se comunicará con todos los demás procesos, actuando como un distribuidor de la información que se quiere mostrar al usuario. Aquí se monta la aplicación BLE que se comentará posteriormente en el epígrafe 5.2.2.
- **Gestión de energía:** este proceso menor, realmente está compuesto por dos programas: el primero se encarga de leer el nivel de batería a través del hardware dedicado mediante I²C, para posteriormente guardar esta información en un archivo que pueda ser consultado por el Servidor BLE; por otra parte, un programa gestionará una interrupción dedicada a escuchar la pulsación que se pueda dar sobre el botón, de tal manera que el resto de procesos podrán comprobar que el usuario quiere apagar el dispositivo, y por tanto finalizarán adecuadamente para posteriormente proceder a un apagado del sistema mediante la salida habilitada para ello.

Es importante resaltar que todos los procesos correrán de manera “concurrente” dentro de las capacidades del procesador, de un solo núcleo. De manera adicional, se puede observar cómo un cuarto proceso se encargará de la gestión de energía. Este proceso a su vez se divide en dos, y está directamente relacionado con el hardware, de tal manera que se encargará de leer las pulsaciones del botón, así como comunicarse con el integrado MAX17263 mediante I²C para monitorizar la batería, y utilizará un mecanismo diferente de comunicación.

Para implementar este mecanismo, se ha desarrollado un *script* en Python que se encargará de inicializar al dispositivo MAX17263 si fuera necesario, y de preguntar cada cierto tiempo el nivel de batería. Este tiempo será inicialmente de un minuto. El valor leído se guardará como porcentaje en un archivo de texto plano. Este mecanismo será sencillo de integrar en el código de Bluetooth, que simplemente tendrá que leer periódicamente el mismo archivo para enviar el nivel de batería al cliente.

Por otra parte, la gestión del apagado mediante el botón se realizará para mejorar el tiempo de vida de los elementos del sistema, y evitar posibles fallos de memoria en aquellos que manejan registros, como ya se comentó en el epígrafe 4.4. De esta manera, existe un proceso encargado de detectar la pulsación del botón mediante una interrupción, escrito en lenguaje C. Este proceso comunicará al *script* principal de Python esta pulsación mediante la escritura en un archivo de texto. Cuando uno de los procesos de Python detecta este cambio en el fichero, propaga una señal que hará que todos los procesos terminen y, por tanto, se haga efectivo el apagado.

5.2.2. Bluetooth Low Energy: Servidor

El desarrollo del periférico o servidor de BLE sobre la Raspberry Pi Zero W está basado principalmente en el código de ejemplo que proporciona BlueZ, pila oficial para el protocolo Bluetooth en Linux [80] [81].

Estos ejemplos muestran cómo implementar los diferentes componentes de la comunicación BLE basada en el protocolo GATT (ver Epígrafe 2.1.2), además del *Advertising* necesario para que la aplicación cliente pueda encontrar y reconocer al dispositivo. El pilar fundamental de este programa es *D-Bus* [82], un sistema que se utiliza en Linux para comunicar procesos con orígenes distintos, es decir, de diferente lenguaje y naturaleza. Hace un trabajo parecido a la librería *multiprocessing* de Python de la que se ha hablado anteriormente, pero a un nivel más bajo, comunicando un gran número de aplicaciones dentro del sistema Linux. Con este elemento, el programa de BLE en Python podrá comunicarse directamente con el software controlador de Bluetooth provisto por BlueZ, y realizar las operaciones pertinentes.

De esta manera, se desarrolla una aplicación como la que se puede ver en la Figura 58.

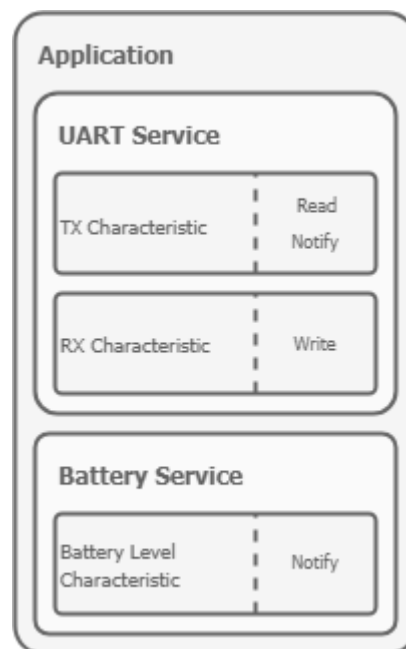


Figura 58. Esquema aplicación cliente BLE.

Se implementan dos servicios:

- **UART Service.** Se desarrolla para poder montar una comunicación que simule una UART y por tanto facilite la comunicación con el cliente. Por un lado, la característica TX se utilizará para enviar los datos obtenidos de ritmo cardíaco y, por otra parte, RX se empleará en la recepción de comandos por parte del usuario (START y STOP).
- **Battery Service.** Como su propio nombre indica, este servicio interactuará con la batería, comunicándose con el programa de gestión energética, mediante el

mecanismo anteriormente comentado, para de esta manera obtener el nivel de batería y poder enviárselo al cliente mediante la característica *Battery Level*.

A la derecha de cada característica aparece el tipo de comunicación que se establecerá con dicha característica, desde el punto de vista del cliente: 1) *write*, para características donde escribirá el cliente; 2) *read*, que en este caso lee el cliente; 3) *notify*, que pueden funcionar a modo de interrupción para el cliente, de tal manera que cuando exista un valor se le notifique al cliente y automáticamente atienda la notificación.

Por otra parte, es importante que Bluetooth ofrece una serie de servicios con un UUID determinado, y unas características prefijadas [83]. Entre estos servicios se encuentra precisamente el de *Heart Rate* que, por sentido común, sería el servicio a implementar en este periférico. Sin embargo, hay una flexibilidad que añade el servicio UART que es la de poder comunicar ambos dispositivos de manera tradicional, enviando y recibiendo datos por dos canales distintos, sin tener que abrir otro servicio para poder enviar datos desde el cliente. Por tanto, como estos servicios prefijados por parte de Bluetooth no son más que una recomendación para acercarse a la estandarización de las comunicaciones en BLE, y en este caso se va a desarrollar una aplicación específica para el dispositivo, se mantendrá el esquema aquí presentado con el servicio UART para el desarrollo de este primer sistema inicial, pudiendo implementarse variaciones en un futuro.

Para realizar pruebas de funcionamiento de este software, se ha utilizado la aplicación *nRF Connect* de *Nordic Semiconductor* [84], que se puede descargar de manera gratuita en la *Play Store* de Android. Esta aplicación permite escanear, descubrir los servicios y comunicarse libremente con todos los periféricos de BLE que sean detectables. A continuación, se presenta una captura de pantalla donde se puede comprobar la implementación ya comentada.

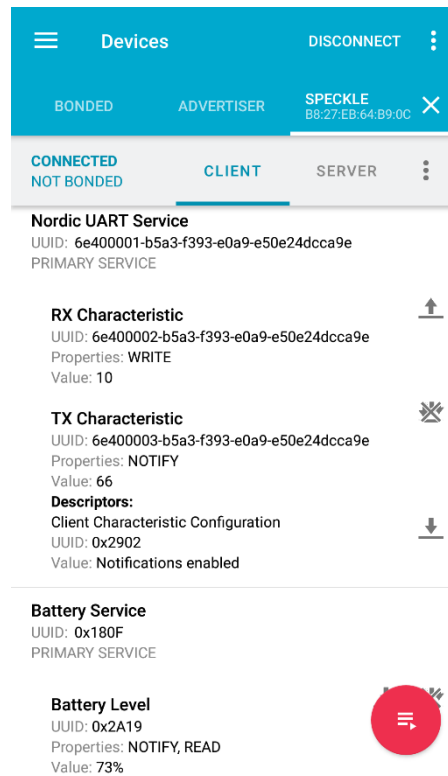


Figura 59. Captura de aplicación nRF Connect de Nordic Semiconductor.

5.2.3. Bluetooth Low Energy: Cliente

Como se ha comentado anteriormente, para la comunicación mediante BLE con el monitor de ritmo cardíaco, se desarrolla una aplicación sobre **Android**, de manera que se tenga un prototipo final donde la comunicación sea robusta. Debido al carácter de prototipo que tiene el sistema, las funcionalidades implementadas serán básicas, definidas a continuación:

- Enviar señales de inicio (START) y parada (STOP) al proceso de monitorización implementado en la Raspberry Pi Zero W.
- Visualizar en todo momento el nivel de batería restante del sistema, así como el valor de ritmo cardíaco cuando la situación lo requiera.

Es importante destacar que las funcionalidades pueden ser objeto de ampliación en un futuro, ya que las aquí presentadas se implementan para poder realizar una prueba de funcionamiento completa del sistema. Las diferentes funcionalidades que se pueden añadir son numerosas y se escapan del objeto del presente trabajo. Por lo tanto, en lo que sigue, se ofrecerá una explicación de las diferentes actividades que se han implementado en la aplicación, y de las librerías utilizadas. Posteriormente, se indicarán diferentes problemas y cuestiones que han aparecido en su desarrollo.

Al abrir la aplicación, se le presenta al usuario una pantalla de introducción donde se pueden observar las diferentes entidades de las que forma parte el proyecto, así como el

propio logotipo del proyecto, diseñado para darle una imagen al mismo, orientado a la aplicación final. En la parte superior aparece un menú donde se dispone del botón para escanear en busca del dispositivo Bluetooth o detener este escaneo, y de un botón de información. Se puede observar esta pantalla en la Figura 60:

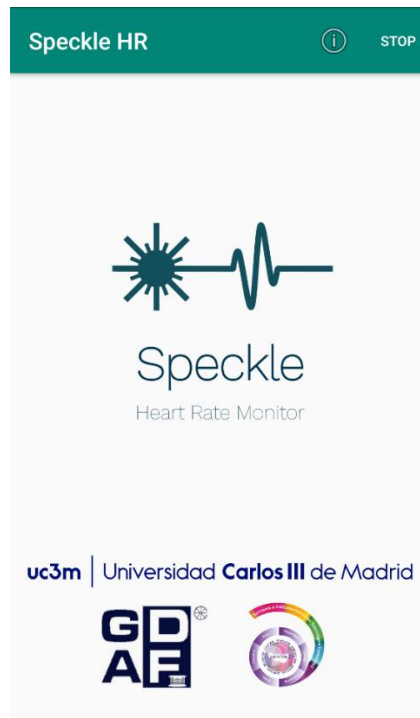


Figura 60. Aplicación cliente en Android. Pantalla de introducción.

El escaneo en busca de dispositivos se realiza automáticamente al abrir la aplicación. En caso de encontrar el dispositivo, la aplicación intentará establecer la conexión con el mismo, de manera que al usuario se le presentará la pantalla principal de interacción. En esta pantalla se dispondrá de los botones destinados a poder comenzar y detener mediciones. Además, el usuario podrá comprobar el valor de ritmo cardíaco a lo largo del tiempo en una gráfica, así como el nivel de batería del dispositivo. En la Figura 61 se puede observar una imagen de esta actividad durante su funcionamiento.

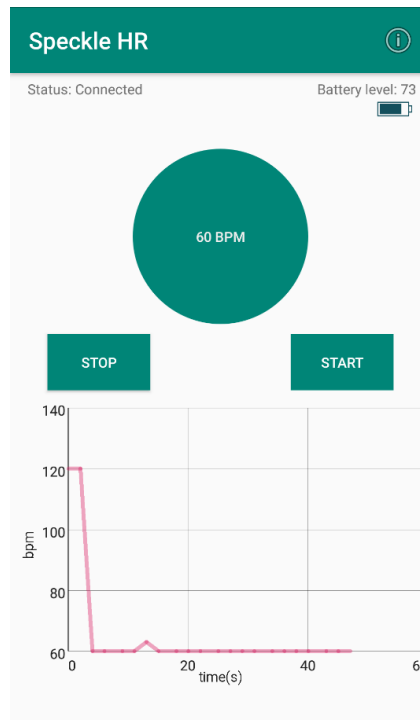


Figura 61. Aplicación cliente en Android. Pantalla de interacción.

Por otra parte, como se ha comentado anteriormente, en el menú superior se dispondrá de un botón de información que permanecerá disponible en todo momento. Este botón abrirá una nueva actividad donde se mostrarán los créditos de la aplicación: creadores, librerías, proyectos de referencia, etc. Esta pantalla se puede consultar en la Figura 62.

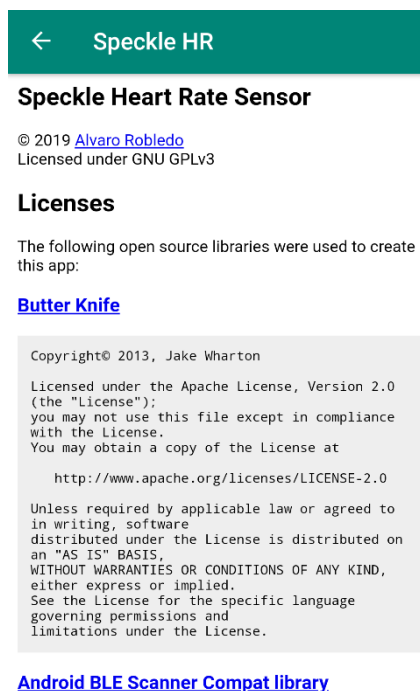


Figura 62. Aplicación cliente en Android. Pantalla de información.

Para el desarrollo de la aplicación, se han empleado algunas librerías con licencia de código abierto, que se detallan a continuación:

- ***Android BLE Scanner Compat Library*** [85]: se trata de una librería desarrollada por la empresa *Nordic Semiconductor*, con el fin de simplificar el escaneo de dispositivos BLE en Android, tarea que ha sido tradicionalmente problemática en el desarrollo de aplicaciones. Además, se proporciona soporte a versiones antiguas de Android que pueden no disponer de algunas funcionalidades implementadas en versiones actual. Para escanear dispositivos se emplearán las funciones proporcionadas por esta librería, mientras que para interactuar con el hardware se utilizarán funciones nativas de BLE en Android.
- ***GraphView*** [86]: esta librería desarrollada por el usuario de GitHub *jjoe64* permite crear diferentes gráficas de manera sencilla, pudiendo darles funcionalidades dinámicas y con un nivel alto de personalización.
- ***Butter Knife*** [87]: en este caso, esta librería está totalmente orientada a simplificar la programación de las vistas y elementos que conforman las diferentes pantallas de la aplicación. Esto incluye tanto su inicialización o visibilidad, como la interacción con las mismas.

Capítulo 6

Medidas y pruebas de integración

Para poder comprobar el funcionamiento de todos los elementos del sistema, tanto de software como de hardware, se realizan integraciones parciales de manera que se vaya comprobando gradualmente el correcto funcionamiento. En cuanto al software, las pruebas se harán con el software de procesado que mejores resultados ha dado hasta el momento, que es el procesado mediante comparación de **intensidad** de los píxeles.

Antes de comprobar el funcionamiento con el chaleco midiendo el ritmo cardíaco de personas, es importante realizar medidas con valores conocidos para poder medir el rendimiento del sistema en un entorno controlado. Para poder medir con el chaleco será necesario implementar sistemas de detección de ruido y de movimientos que permitan filtrar la información no deseada. Por lo tanto, se utilizará un zumbador electrónico como dispositivo de ensayo (DUT) que permita facilitar la posterior depuración del software, tal y como se comenta previamente en el epígrafe 3.2. Para realizar medidas, se introduce luz en la fibra mediante uno de los láseres vistos en el apartado 4.2.2 (aunque todas las medidas aquí mostradas se han realizado con el láser de LasIRvis, ver Figura 24), haciendo pasar la fibra por una caja donde se encuentra el zumbador, sujeto entre almohadillas para amortiguar cualquier perturbación externa (Figura 63).



Figura 63. Caja con zumbador electrónico.

Posteriormente, se recibe la señal de *speckle* resultante con la cámara, tal y como se muestra en la Figura 64. Es importante destacar que la salida de la fibra incidirá

directamente en el sensor CMOS de la cámara, puesto que no se han encontrado referencias en la bibliografía a la necesidad de acoplar ópticamente la salida de la fibra al sensor.

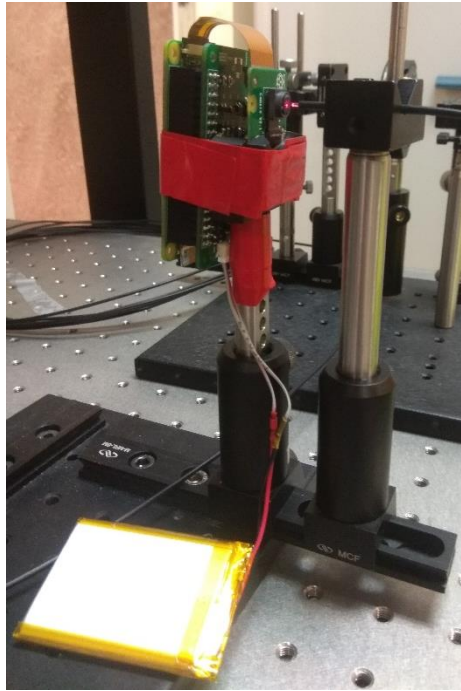


Figura 64. Set-up de la recepción en el montaje para las pruebas.

Las pruebas realizadas son las siguientes:

- 1. Pruebas de software.** Estas pruebas se realizan para comprobar el correcto funcionamiento del procesado. Para ello, se realiza un montaje en una mesa óptica para acoplar la fibra tanto al láser como a la cámara, y la Raspberry se alimenta mediante un adaptador de red. Se toman varias medidas a cada frecuencia para poder tener después una tabla y calcular los errores. Los datos se obtienen de manera remota mediante terminal (SSH, *Secure Shell*).
Objetivos: caracterizar el error, saber en qué frecuencias funciona mejor y peor.
- 2. Pruebas de hardware.** Para esta prueba se integra la batería, así como el circuito diseñado. Aquí simplemente se comprueba que el hardware funciona correctamente. Esto incluye también pruebas de la parte de encendido y apagado.
Objetivos: monitorizar valor de batería, encendido y apagado correcto.
- 3. Pruebas de Bluetooth e integración software.** En este caso se solapa un poco con la prueba anterior, de manera que ahora interesa que se pueda iniciar y parar el procesamiento mediante la aplicación, así como que se pueda recibir la información enviada mediante Bluetooth. Para esto, los programas deben iniciarse automáticamente.
Objetivos: recibir ritmo cardíaco y batería, enviar START y STOP.
- 4. Pruebas de sistema completo.** Finalmente se realiza una prueba final para realizar una última comprobación del funcionamiento del sistema en su conjunto.

Se varía la frecuencia de excitación del zumbador en tiempo real para comprobar la respuesta del sistema implementado.

Objetivos: funcionamiento completo del sistema.

La planificación de las pruebas se muestra en la Tabla 5 y es la siguiente:

Aparecen del 1 al 3 las pruebas a realizar, descritas previamente, y en verde están los días estimados en los que estarían las pruebas realizadas y/o en proceso de ejecución. Arriba aparecen los días del mes de mayo en los que se realizan las pruebas, junto a sus respectivos días de la semana.

Tabla 5. Plan de pruebas de integración.

	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V
	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1																	
2																	
3																	
4																	

A continuación, se detallarán las observaciones realizadas con las diferentes pruebas de integración, así como los procedimientos seguidos para implementar la integración en cuestión.

1. Pruebas de software.

Mediante esta prueba, siguiendo el procedimiento anteriormente comentado, se realizan medidas con un zumbador acoplado a la fibra, para posteriormente comparar los resultados obtenidos con las frecuencias de excitación aplicadas. Con el fin de aproximarse lo más posible a una situación real, el zumbador se excitará con la señal de amplitud más pequeña posible para su funcionamiento. Las medidas se realizan en intervalos temporales de un minuto, aplicando una frecuencia constante en el zumbador, y comparando posteriormente la media obtenida de las medidas a lo largo del intervalo con la frecuencia aplicada. Aunque se utilizará el procesamiento mediante intensidad, el filtrado y el análisis frecuencial se realizará en cada una de sus variantes, para poder ver cómo se comporta cada uno de ellos.

Es importante destacar en este punto la participación de Ignacio Sánchez, quien se encarga tanto de tomar las medidas realizadas como de realizar después un análisis de los errores. Los resultados que se obtienen mediante este análisis se pueden consultar en la Figura 65, que muestra medidas realizadas con una amplitud de excitación del zumbador de 3,5 V, aunque como se ha comentado anteriormente, las pruebas se han realizado a diferentes amplitudes de excitación.

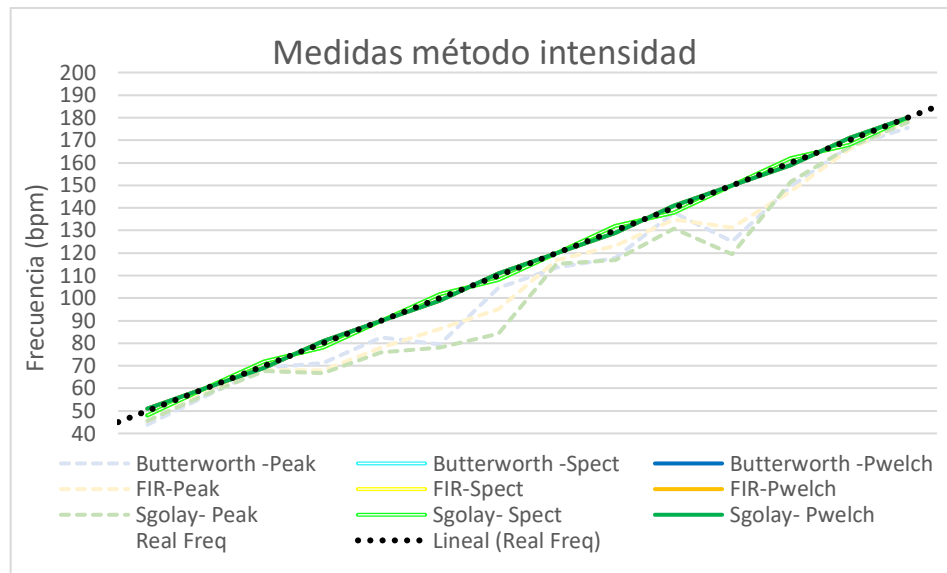


Figura 65. Medidas usando diferentes procesados con método de intensidad a 3,5 V.

Como se puede comprobar, las medidas con los análisis *findpeaks* que analizan la señal en busca de máximos locales (líneas discontinuas) presentan desviaciones importantes respecto al valor real, lo cual se ha comprobado que ocurre también con el filtrado Savitzky-Golay (color verde) para amplitudes más bajas. Sin embargo, los filtrados Butterworth (color azul) y FIR (color amarillo), junto a los análisis mediante espectrograma (líneas claras) y el método de Welch (líneas oscuras) son considerablemente precisos en todo el rango, teniendo este último ligeramente mejores resultados.

2. Pruebas de hardware

En estas pruebas, en primer lugar, se comprobará que los valores proporcionados por el monitor de **batería** son correctos. Para inicializar dicho componente se utiliza la guía que ofrece *Maxim Integrated* para la familia MAX1726X [88]. En dicho documento, se establecen los tres valores necesarios para que el algoritmo *ModelGauge* m5 EZ pueda comenzar a funcionar. Estos valores son la capacidad nominal de la batería, la corriente de final de carga y el voltaje de vaciado.

Mediante la terminal de la Raspberry Pi, se hace uso de los comandos *i2cdetect*, *i2cset* e *i2cget* con el fin de inicializar valores sobre la marcha sin necesidad de implementar un programa dedicado. Para ello se sigue la guía mencionada, escribiendo y leyendo los diferentes registros.

Los principales problemas encontrados y posteriormente solventados son los siguientes:

- Al no poder leer medidas del nivel de batería, pero sí medidas de corriente mediante la resistencia de sensado (Figura 35), se revisa la circuitería para descubrir un error en el conexionado del integrado: el pin CSPH (Figura 35),

necesario para la medida de voltaje de la batería, se diseña como un pin flotante, en vez de estar conectado a la línea positiva de la alimentación. Por tanto, se realiza una modificación sobre el circuito implementado, como se puede comprobar en la Figura 66. Tras resolver este error en el diseño, se obtienen medidas del nivel de batería.

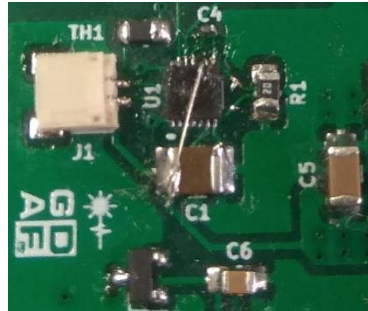


Figura 66. Solución al error en el conexionado del integrado MAX17263.

- Se observan errores de medida en el nivel de batería a lo largo de diferentes ciclos de carga y descarga, llegando a mostrar un valor de 0% durante varias horas con el sistema encendido. Esto se debe al valor escogido de la resistencia de medida de corriente del contador de coulomb. Como se puede observar en la Tabla 6, para el valor de resistencia diseñado de 2 m Ω , se tiene una resolución de 2,5 mAh, lo que significaría un rango de $\pm 25,6$ A. Esto está, evidentemente, muy por encima de los rangos que se obtendrán en la aplicación final, siendo la máxima corriente a medir de 0,5 A durante la carga. Esto supone que los valores de corriente a medir utilicen una parte muy reducida del rango total disponible, lo que puede dar lugar a errores debido a la resolución utilizada. Por lo tanto, se decide optar por un valor de resistencia de 20 m Ω , con el fin de tener una resolución menor y un rango que se pueda aprovechar mejor. Los valores de nivel de batería observados tras esta modificación son lógicos a lo largo de varios ciclos, por lo que este apartado quedaría solventado.

Tabla 6. Rangos de corriente y resolución en función del valor de resistencia [64].

BATTERY FULL CAPACITY (mAh)	SENSE RESISTOR (m Ω)	CURRENT REGISTER RESOLUTION (μ A)	CURRENT REGISTER RANGE (A)	CAPACITY RESOLUTION (mAh)
> 4000	1	1562.5	± 51.2	5
> 2000	2	781.25	± 25.6	2.5
> 800	5	312.5	± 10.24	1
> 400	10	156.25	± 5.12	0.5
> 200	20	78.125	± 2.56	0.25
> 80	50	31.25	± 1.024	0.1

Por otra parte, respecto al circuito de **encendido y apagado**, se observa un funcionamiento óptimo en la mayoría de los casos, habiendo situaciones en las que el biestable implementado vuelve al estado anterior tras pulsar el botón. Se estima que ésto es debido a la carga y descarga de algún condensador. Sin embargo, se han realizado modificaciones sobre el circuito implementado sin observar un

funcionamiento correcto en el 100% de los casos. Por lo tanto, de cara a trabajos futuros, sería necesario revisar este apartado del circuito, e incluso abrir la posibilidad de aproximar la solución desde un enfoque distinto, manteniendo el concepto de diseño.

3. Pruebas de Bluetooth e integración software

En este caso, el sistema final es testeado a la hora de realizar medidas teniendo como única interfaz de usuario la propia aplicación de Android. Para que esto pueda tener lugar, es necesario que todos los scripts programados se ejecuten automáticamente, y que sean los encargados de gestionar el apagado del sistema cuando así se requiera. Al tratarse de un sistema operativo Linux, esto es sencillo de realizar mediante el lanzamiento de los programas desde el archivo *rc.local*, script normalmente utilizado para este propósito, que se ejecuta durante el encendido o *boot* del sistema. Por lo tanto, al encender la Raspberry Pi Zero W, la aplicación desde el *smartphone* debe ser capaz de buscar y conectarse con el hardware.

Los principales problemas encontrados en este apartado son principalmente debidos a fallos de conexión desde la aplicación. Mientras que con la aplicación *nRF Connect* (apartado 5.2.2) la conexión y la comunicación se realiza sin ningún problema, con la aplicación desarrollada en ocasiones se necesitan varios intentos para establecer una conexión robusta. Presumiblemente, se cree que el motivo de esto puede venir dado por un error de sincronización de procedimientos en el manejo de la API de Bluetooth para Android.

A pesar de este problema, se observa que la comunicación funciona correctamente, pudiendo comprobar el desempeño de todas las funcionalidades implementadas, además de recibir en todo momento información de la batería o del valor de medida del sensor en latidos por minuto (bpm, *beats per minute*) si la situación lo requiere.

4. Pruebas de sistema completo

Para comprobar el funcionamiento del sistema en su versión completa, se realizan medidas variando la frecuencia de excitación del zumbador, con el objetivo de observar estas variaciones desde la aplicación. El *setup* de la presente prueba se puede observar en la Figura 67, donde se encuentran los diferentes elementos del sistema. El láser directamente acoplado a la fibra se encuentra alimentado por una fuente de tensión, y el zumbador dentro de la caja se excita mediante un generador de funciones. Tanto la Raspberry Pi como la electrónica desarrollada para el presente trabajo se encuentran alimentados mediante la batería. De cara al sistema final, y como se ha implementado en el hardware, la Raspberry Pi debería ser capaz de alimentar el mismo laser mediante una salida digital, tal y como se ha habilitado.

No obstante, al no disponer del láser de LasIRvis (Figura 24) comentado en el apartado 4.2.2, se ha tenido que utilizar el láser de la Figura 27 del mismo apartado. Éste láser se alimenta con 5V, y habría que modificar la electrónica del módulo para que pudiese funcionar con los 3,3V de la Raspberry Pi. Como se observa en la imagen, el sistema implementado estará colocado de tal manera que la cámara pueda quedar enfrentada al extremo de la fibra, para detectar el patrón de *speckle*.

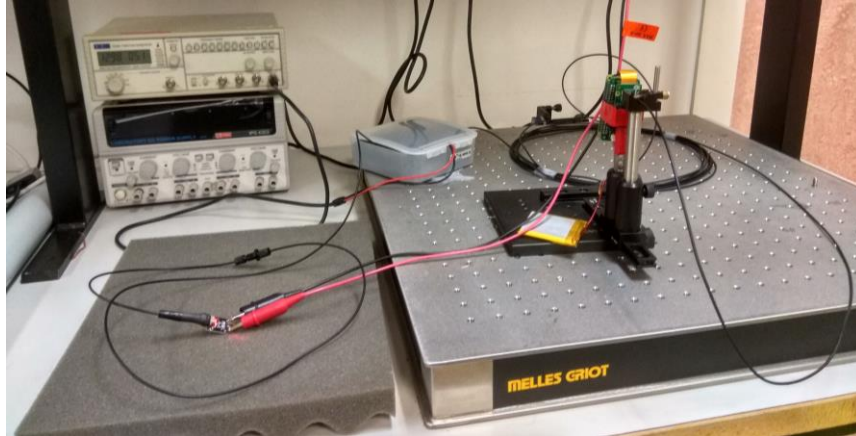


Figura 67. Setup para medidas con sistema completo.

Como se observa en la Figura 68, el sistema responde correctamente ante las variaciones en la frecuencia del zumbador. Adicionalmente, se puede comprobar cómo el nivel de batería (en porcentaje) ha disminuido a lo largo de las mediciones. El sistema presente presenta la robustez de poder estar trabajando de manera continua períodos extensos de tiempo. Como se muestra en la imagen, las variaciones de frecuencia se introducen en torno a los 7 minutos de funcionamiento.

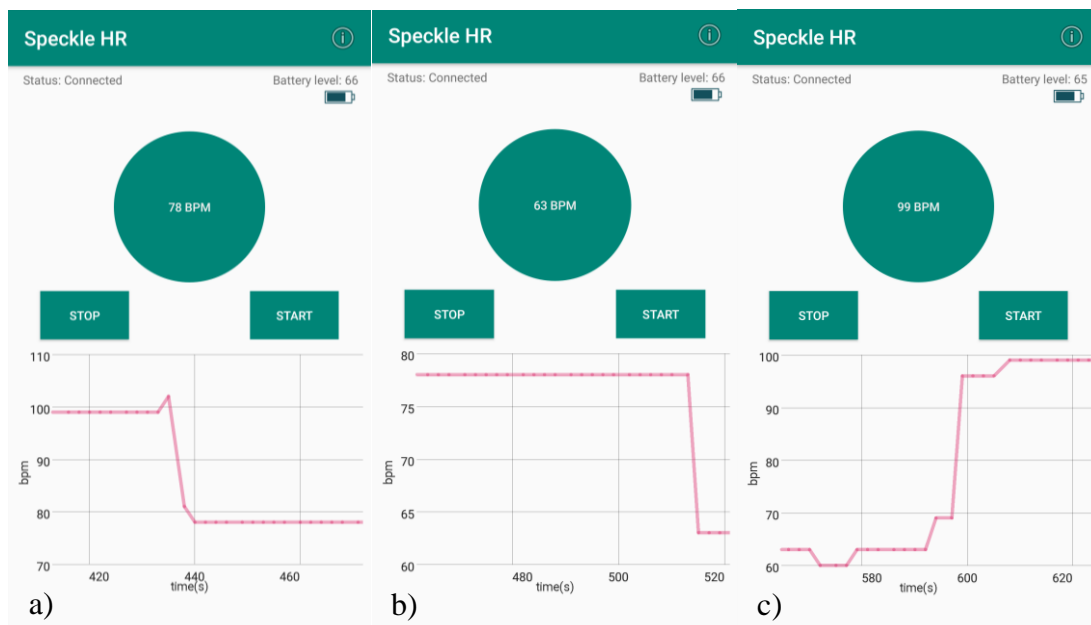


Figura 68. Medidas con variación de frecuencia de excitación. a) variación de 1,647 Hz a 1,298 Hz; b) variación de 1,298 Hz a 1,044 Hz; c) variación de 1,044 Hz a 1,650 Hz, y cambio de nivel de batería de 66% a 65%. El eje vertical se muestra en bpm, de tal manera que 60bpm = 1 Hz.

Finalmente, tras las pruebas realizadas, se puede resumir que el sistema cumple con las siguientes especificaciones:

- Sistema basado en microprocesador de 1 GHz.
- Sistema operativo Raspbian basado en Linux.
- Comunicación Bluetooth Low Energy.
- Capacidad de batería de 2000 mAh.
- Consumo medio de 300 mA.
- Operación continua teórica de 6h 30min.
- Peso: 67 g.
- Dimensiones sin carcasa: 43,5 x 76,7 x 19,6 mm.
- Dimensiones carcasa: 46 x 100 x 39 mm.

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusiones

En este trabajo se presenta una técnica novedosa y no intrusiva para monitorizar el ritmo cardíaco, desarrollando una integración de múltiples componentes para implementar un dispositivo *wearable* y realizando una serie de medidas con un dispositivo de ensayo que emula el ritmo cardíaco, demostrando su correcto funcionamiento. La técnica de *speckle* para monitorizar constantes vitales se ha presentado por tanto como una alternativa para situaciones en las que la radiación electromagnética es elevada dificultando el uso de otros sistemas.

Se ha realizado un diseño a nivel de *hardware* de una placa de expansión para la Raspberry Pi Zero W, implementando un sistema de gestión energética para poder cargar una batería y monitorizarla mediante I²C; y un sistema para poder gestionar el apagado y encendido del dispositivo. Además, se ha llevado a cabo una completa simplificación del bloque óptico implementado en el sistema previo, con una nueva fuente de luz, tras la cual se han podido tomar medidas correctas.

A nivel del hardware, solamente han surgido problemas no totalmente resueltos en el circuito lógico, donde se observan comportamientos inesperados que han sido analizados en detalle sin poder extraer una solución definitiva.

A nivel de *software*, se han implementado una serie de *scripts* para poder procesar en tiempo real el patrón de *speckle*, portador de la señal de ritmo cardíaco; y un sistema para comunicar el resultado obtenido de manera inalámbrica con BLE, utilizando como cliente una aplicación sobre Android.

Las técnicas de procesamiento propuestas son numerosas y se ha analizado el rendimiento de cada una, teniendo un resultado global positivo.

Todo esto se ha realizado sobre una plataforma de software libre como es Raspbian, utilizando también diferentes librerías *open source* para su programación.

Por último, se ha propuesto un diseño de una carcasa 3D para poder integrar el sistema completo, que sirve para mostrar la viabilidad de la implementación del sistema *wearable*.

Por lo tanto, se considera que los objetivos inicialmente planteados se han cubierto por completo, obteniendo como resultado un trabajo completo, multidisciplinar y de considerable valor técnico.

7.2. Líneas futuras

Parece razonable que los futuros trabajos que se realizarán en este tema versarán sobre la implementación de lo desarrollado en la aplicación final.

En primer lugar, será necesario continuar desarrollando esfuerzos sobre el prototipo implementado, para poder realizar pruebas concluyentes sobre su utilización con personas. Se proponen los siguientes apartados a desarrollar:

- El procesamiento deberá seguir una línea de mejora y optimización para poder lograr la mayor eficacia posible. Para esto, se propone implementar técnicas avanzadas de reducción de ruido, así como una revisión del procesamiento en busca de cuellos de botella que puedan limitar las capacidades del sistema.
- Una vez establecidos los métodos más eficaces de procesamiento en tiempo real, el software implementado sobre Raspbian podrá implementarse en una plataforma *Embedded Linux* dedicada al desarrollo de sistemas empotrados, con el fin de optimizar los recursos disponibles del procesador, ya que se reduciría una gran cantidad de procesos internos que se llevan a cabo dentro de un sistema operativo convencional.

Por otra parte, cuando quede patente el correcto funcionamiento del sistema en su aplicación y se establezcan las pautas a seguir para un procesamiento eficaz, se podrá ir más allá y empotrar el sistema en un dispositivo dedicado, tanto reduciendo el tamaño del hardware como implementando tecnologías novedosas como son la óptica y la electrónica flexible.

Otra línea de desarrollo relacionada con el presente trabajo puede ser la implementación de un sistema estático, donde las limitaciones de tamaño y de energía serían menores, y por tanto el procesamiento podría trabajar con mayor soltura.

Por último, con el fin de desarrollar un dispositivo totalmente accesible, se propone, en la línea del desarrollo de la aplicación Android, implementar mayores funcionalidades para mejorar la experiencia del usuario.

Capítulo 8

Entorno socioeconómico

En este capítulo se enmarcará el proyecto en el ámbito socioeconómico, presentando inicialmente los costes de realización del mismo, para después indicar las implicaciones tanto sociales como económicas que tienen proyectos de estas características.

8.1. Presupuesto

A continuación, se presentan los costes de realización del proyecto (Tabla 7), el cual, como se ha comentado inicialmente, ha sido cofinanciado por los programas Sensores e Instrumentación en Tecnologías Fotónicas (SINFOTON-CM, Ref: S2013/MIT-2790), SINFOTON2-CM Ref: S2018/NMT-4326 y GREENFIBER Ref: TEC2015-63826-C3-2-R. El presupuesto se ha dividido en tres capítulos: equipos y software, costes materiales y recursos humanos.

Además, se han tenido en cuenta varios aspectos generales:

- La amortización de los equipos y el software es lineal con valor residual nulo. La duración de trabajo y, por tanto, la base de los cálculos de amortización es de 6 meses. Además, se estiman 5 años de vida útil para todos los equipos.
- En la línea de lo anterior, el coste de los recursos humanos se tomará de acuerdo a un período de 6 meses a jornada parcial. Esto supone por lo tanto un número de 480 horas.
- Se ha utilizado *software* de código abierto (*KiCad*, *FreeCAD* y librerías), que por tanto es de uso libre y gratuito.
- En las tablas: “U.M.” es Unidad de Medida; “P.U.” es Precio Unitario; y “P.T.” es Precio Total. Se aplican los gastos de transporte. No se aplican otros gastos de gestión ni los impuestos territoriales correspondientes, así como otros costes indirectos.

Tabla 7. Presupuesto del proyecto desglosado por capítulos.

CÓDIGO	U.M.	DESCRIPCIÓN	MEDICIÓN	P.U.	P.T.
CAPÍTULO 1: EQUIPOS Y SOFTWARE					
1.01	unidad	Osciloscopio digital RIGOL DS1062C. <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 717,14 €.</i>	1	71,71 €	71,71 €
1.02	unidad	Fuente de tensión IPS 4303 (RS PRO) <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 358,00 €.</i>	1	35,80 €	35,80 €
1.03	unidad	Medidor Poptica Thorlabs PM20A <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 511,88 €.</i>	1	51,19 €	51,19 €
1.04	unidad	Estación de soldadura ANTEX 660-TC <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 189,83 €.</i>	1	18,98 €	18,98 €
1.05	unidad	Multímetro digital RS-14 (RS PRO) <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 27,83 €.</i>	1	2,78 €	2,78 €
1.06	unidad	Ordenador personal HP Pavilion 15-e043ss. Sistema operativo Windows 10, procesador Intel Core i5-3, 512 GB de almacenamiento SSD y 8 GB de memoria RAM. <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 699 €.</i>	1	69,90 €	69,90 €
1.07	unidad	Mesa óptica MELLES GRIOT. <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 848 €.</i>	1	84,80 €	84,80 €
1.08	licencia	MATLAB R2018b. Licencia académica de uso personal. <i>Amortización lineal con duración de 1 año y precio de adquisición de 250 €</i>	1	125,00 €	125,00 €
			TOTAL. CAPÍTULO 1.	460,17 €	

CAPÍTULO 2: COSTES MATERIALES

2.01	unidad	Raspberry Pi Zero W. Procesador Broadcom BCM 2835, RAM 512 MB, WiFi, Bluetooth 4.1 (con BLE), puerto USB-OTG, conector de cámara CSI. Kit con carcasa. Envío incluido.	1	42,14 €	42,14 €
------	--------	---	---	---------	---------

2.02	unidad	Pi Camera v2. Sensor de imagen Sony IMX219 de 8MP, resolución hasta 3280x2464 píxeles, conector CSI. Envío incluido.	1	43,92 €	43,92 €
2.03	unidad	PCB fabricada por PCBWay. Pedido de coste mínimo de 10 PCBs. Placas con espesor 1,6 mm y doble capa, color verde. Envío incluido.	10	2,12 €	21,18 €
2.04	unidad	Componentes electrónicos. Incluye componentes pasivos (resistencias y condensadores), transistores P-MOSFET, diodos (Schottky, TVS, LED), componentes lógicos (biestable, puerta OR), integrados de carga y monitorización de batería, conectores y pulsador. Entre dos y tres elementos por referencia, según criticidad. Envío incluido.	1	56,56 €	56,56 €
2.05	metros	Fibra óptica de plástico (POF) de PMMA	10	0,52 €	5,20 €
2.06	unidad	Diodo láser. Módulo láser de Akozon. 5mW a 650nm. Envío incluido.	4	2,25 €	8,98 €
TOTAL. CAPÍTULO 2.					177,98 €

CAPÍTULO 3: RECURSOS HUMANOS

3.01	hora	Ingeniero técnico industrial con especialidad en electrónica. Nivel de experiencia profesional mínimo junior.	15,222200		
		<i>Incluidos gastos de seguridad social e indemnización.</i>	480	€	7.200,00 €
TOTAL. CAPÍTULO 3.					7.200,00 €

TOTAL. 7.838,15 €

Por tanto, se puede realizar un resumen de los costes por capítulo, que se muestra en la Tabla 8:

Tabla 8. Resumen presupuesto por capítulos.

CAPÍTULO	DESCRIPCIÓN	P.T.
1	EQUIPOS Y SOFTWARE	460,17 €
2	COSTES MATERIALES	177,98 €
3	RECURSOS HUMANOS	7.200,00 €
TOTAL		7.838,15 €

El presupuesto total de este proyecto asciende a la cantidad de **SIETE MIL OCHOCIENTOS TREINTA Y OCHO CON QUINCE EUROS.**

8.2. Impacto socioeconómico

El presente trabajo presenta algunas implicaciones, tanto sociales como económicas que conviene resaltar en este epígrafe. Hay que destacar inicialmente que el prototipo diseñado está orientado al desarrollo de un dispositivo *wearable*, principalmente orientado a su aplicación en la atención médica.

8.2.1. Impacto social

Los dispositivos *wearables* se están introduciendo de manera progresiva en la atención médica, basándose en el principio de personalización [89]. Debido a factores como el creciente número de personas y el aumento de la edad media de la población, los *wearables* de estas características suponen una solución para que los médicos puedan monitorizar a los pacientes mientras se encuentran en sus hogares, lo que permite liberar habitaciones en los hospitales tan concurridos de la actualidad. Los *wearables* también pueden ser útiles para prevenir enfermedades, como afecciones cardiovasculares, causa del 31% de las muertes en el mundo [2].

La creciente conciencia sobre la salud ha traído los años con más desarrollo en sensores y dispositivos que monitorizan la información biométrica de las personas. Desde pacientes hasta trabajadores como policías, bomberos u operarios, incluyendo el interés que supone para el controlar el estilo de vida de la persona promedio; la gente está más preocupada por su bienestar que nunca, y se han desarrollado dispositivos complejos como el marcapasos o el Desfibrilador-Cardioversor Implantable que puede actuar en situaciones de emergencia [90], o aparatos más simples y menos invasivos, como los conocidos relojes inteligentes o *smartwatches*. El sistema aquí presentado trata de implementar un dispositivo que moleste lo menos posible al usuario que lo lleva.

Uno de los principios de los wearables para el cuidado de la salud es el de comercializar los dispositivos de medición clásicos para que se puedan usar a diario, lo que explica la creciente desaparición de los molestos cables y el desarrollo de la comunicación inalámbrica centrada en la conectividad de los wearables, siendo el mejor ejemplo el Bluetooth de baja energía o BLE que se ha utilizado en el trabajo.

Sin embargo, hay dos problemas que preocupan a los usuarios y desarrolladores: para los primeros, la seguridad de sus datos, y para los últimos, el gran volumen de datos a procesar y transmitir [91]. La primera se refiere al procesamiento y almacenamiento seguros de los datos del usuario, siendo necesaria la intervención de las autoridades para evitar el intercambio de datos entre diferentes compañías o el uso de esta información para el beneficio de otro agente. La cuestión del elevado volumen de datos concierne a las compañías, dado que los wearables están constantemente compartiendo información que aumenta exponencialmente. La tecnología aún tiene los desafíos de administrar esta información de manera rápida y eficaz y proporcionar protocolos seguros para la comunicación. Se estima que la próxima generación de comunicaciones 5G establecerá los medios necesarios para la correcta operación de dispositivos IoT como son los *wearables*.

8.2.2. Impacto económico

Según el portal Statista, se estima que el mercado de la salud digital aumentará su valor desde los \$79 millones en 2015 hasta los \$206 millones en 2020, con una tasa de crecimiento (CAGR, *Compound Annual Growth Rate*) del 21%. Por lo tanto, el dispositivo aquí desarrollado se encuentra dentro de un mercado con un crecimiento importante en la actualidad. Concretamente, se estima que el mercado de la salud móvil o *mHealth*, donde se encontrarían los dispositivos *wearables*, va a experimentar un CAGR del 41% entre los años 2015 y 2020 (Figura 69), pudiendo llegar a abarcar un mercado de \$332,7 millones en 2025 [92].

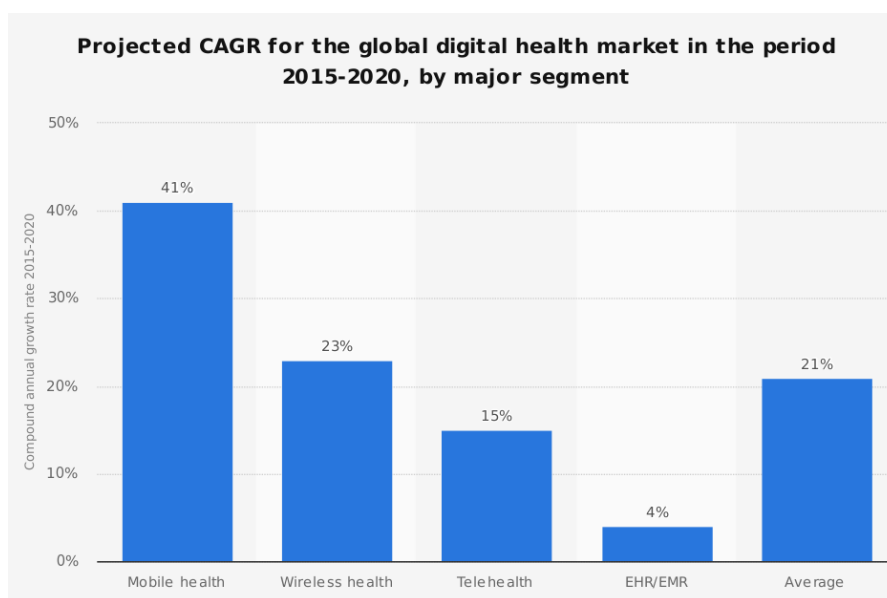


Figura 69. CAGR del mercado de la salud digital según segmento 2015-2020 [92].

Sin embargo, aunque el impacto de dispositivos como el aquí desarrollado sea tan prometedor a nivel internacional, es importante destacar que, aunque en España se dedicara un 9% del Producto Interior Bruto (PIB) a la atención médica en 2016, valor que está en la media de los países de la Organización para la Cooperación y el Desarrollo Económicos (OCDE), el gasto *per capita* está muy por debajo de la media de la OCDE: un 40% menos. Además, en España la financiación aportada por el sector privado es relativamente mayor que en otros países europeos como Alemania, Francia o Reino Unido [93].

Por lo tanto, el sistema desarrollado en el presente trabajo se encuentra enmarcado dentro de un segmento del mercado en continua mejora y crecimiento, que aspira a llegar a su madurez dentro de los próximos 10 años.

Capítulo 9

Planificación global

En este capítulo se pretende dar una visión de la planificación llevada a cabo a lo largo de todo el trabajo. Destaca una parte de aprendizaje inicial, utilizando el sistema previo. De la misma manera, es destacable que el desarrollo del software se ha llevado prácticamente en paralelo con el diseño de la PCB, así como la continua documentación de la presente memoria. Tras esto, se dedica cierto tiempo a las pruebas y cuestiones de integración.

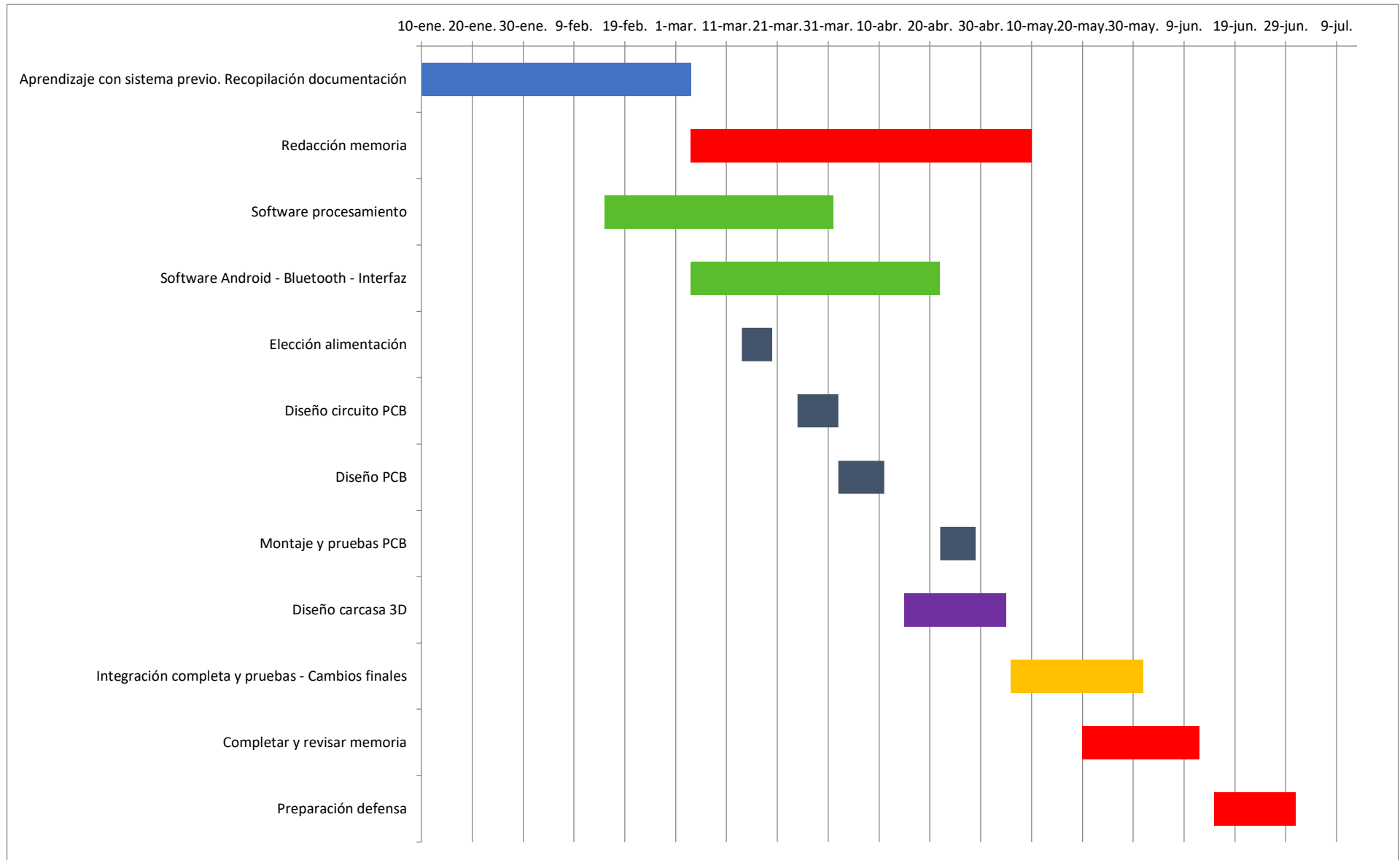


Figura 70. Planificación global de tareas.

Bibliografía

- [1] P. J. Pinzón, D. S. Montero, A. Tapetado y C. Vázquez, «Unobtrusive monitoring of heart rate using a cost-effective speckle-based SI-POF remote sensor,» *Dynamics and Fluctuations in Biomedical Photonics XIV*, vol. 10063, 2017.
- [2] Organización Mundial de la Salud, «Enfermedades cardiovasculares,» 17 Mayo 2017. [En línea]. Available: [https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)). [Último acceso: Abril 2019].
- [3] Organización Médica Colegial de España, «Encuesta sobre la situación del Médico de Atención Primaria en España en 2015 y las consecuencias de los recortes,» 2015. [En línea]. Available: https://www.cgcom.es/sites/default/files/u183/encuesta_ap_resultados_ccaa.pdf. [Último acceso: Abril 2019].
- [4] Parlamento Europeo, *Reglamento (UE) 2017/745 del Parlamento Europeo y del Consejo, de 5 de abril de 2017.*, 2017.
- [5] Parlamento Europeo, *Reglamento (UE) 2017/746 del Parlamento Europeo y del Consejo, de 5 de abril de 2017.*, 2017.
- [6] Boletín Oficial del Estado, «Real Decreto 486/2010, de 23 de abril, sobre la protección de la salud y la seguridad de los trabajadores contra los riesgos relacionados con la exposición a radiaciones ópticas artificiales.,» 27 Abril 2010. [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2010-6485>.
- [7] BOE, «BOE - Código de Propiedad Intelectual,» [En línea]. Available: <https://www.boe.es/legislacion/codigos/codigo.php?id=87&modo=1¬a=0>.
- [8] BOE, «Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual,» [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-1996-8930>.
- [9] Universidad Carlos III de Madrid - Biblioteca, «Propiedad intelectual y derechos de autor: cuestiones generales,» [En línea]. Available: https://www.uc3m.es/ss/Satellite/Biblioteca/es/TextoDosColumnas/1371213999139/Propiedad_intelectual_y_derechos_de_autor:_cuestiones_generales.
- [10] Creative Commons, «When we share, everyone wins - Creative Commons,» [En línea]. Available: <https://creativecommons.org/>.
- [11] Proyecto GNU, «Licencias,» [En línea]. Available: <https://www.gnu.org/licenses/licenses.es.html>.

- [12] J. Desjardins, «Infographic: The History of Wearable Technology,» 20 Mayo 2015. [En línea]. Available: <https://www.visualcapitalist.com/the-history-of-wearable-technology/>. [Último acceso: Septiembre 2018].
- [13] N. M. y. D. H. Gobinath Aroganam, «Review on Wearable Technology Sensors Used in,» *MDPI Sensors*, vol. 19, n° 1983, pp. 1-26, 12 Abril 2019.
- [14] S. Jain, «Imagining 5G: The Future Network of "Things". The Evolving Landscape of IoT and Wearables,» Abril 2017. [En línea]. Available: <https://blogs.cisco.com/sp/imagining-5g-the-future-network-of-things-the-evolving-landscape-of-iot-and-wearables>. [Último acceso: Octubre 2018].
- [15] M. Çiçek, «Wearable Technologies and its Future Applications,» *International Journal of Electrical, Electronics and Data Communication*, vol. 3, n° 4, pp. 45-50, 2015.
- [16] Battery University, «What's the Best Battery?,» 2017. [En línea]. Available: https://batteryuniversity.com/learn/archive/whats_the_best_battery.
- [17] epectec, «Battery Cell Comparision,» [En línea]. Available: <https://www.epectec.com/batteries/cell-comparison.html>.
- [18] Mouser Electronics, «Extending theBattery Life of Wearable Devices,» [En línea]. Available: <https://eu.mouser.com/applications/battery-life-wearable-devices/>.
- [19] Renesas Electronics America, «White Paper - Battery Management System Tutorial,» 2018.
- [20] Battery University, «BU-903: How to Measure State-of-charge,» Octubre 2017. [En línea]. Available: https://batteryuniversity.com/learn/article/how_to_measure_state_of_charge.
- [21] Adafruit Industries, «FLORA - Wearable electronic platform: Arduino-compatible - v3,» [En línea]. Available: <https://www.adafruit.com/product/659>.
- [22] Maxim Integrated, «Ultra-Low-Power 32-bit Microcotrollers,» [En línea]. Available: <https://www.maximintegrated.com/en/products/microcontrollers/low-power-microcontrollers.html>.
- [23] Raspberry Pi, «Raspberry Pi Zero,» [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-zero/>.
- [24] Particle.io, «How to Select a MCU for your IoT Devices,» [En línea]. Available: <https://www.particle.io/mcu-vs-soc-vs-microprocessor-for-iot/>.
- [25] K. Townsend, C. Cufí, Akiba y R. Davidson, *Getting Started with Bluetooth Low Energy*, Sebastopol, CA: O'Reilly, 2014.
- [26] UN DESA, «Statista: Projected distribution of the world's population from 2017 to 2100, by age group,» 2017. [En línea]. Available: <https://www-statista-com.biblioteca5.uc3m.es/statistics/672546/projected-world-population-distribution-by-age-group/>.
- [27] UN DESA, «Statista: Forecast about the development of the world population from 2015 to 2100 (in billions),» [En línea]. Available: <https://www-statista->

com.biblioteca5.uc3m.es/statistics/262618/forecast-about-the-development-of-the-world-population/ .

- [28] J. J. Rutherford, «Wearable Technology: Health-Care Solutions for a Growing Global Population,» *IEEE Engineering in Medicine and Biology Magazine*, pp. 19-24, 2010.
- [29] A. Tognetti, F. Lorussi, R. Bartalesi, S. Quaglini, M. Tesconi, G. Zupone y D. De Rossi, «Wearable kinesthetic system for capturing and classifying upper limb gesture in post-stroke rehabilitation,» *Journal of NeuroEngineering and Rehabilitation*, vol. 2, nº 8, pp. 1-16, 2005.
- [30] O. Peltola, *Introduction to Wearable Healthcare Technology*, Jyväskylä: University of Jyväskylä, 2017.
- [31] D. T. Weiler, S. O. Villajuan, L. Edkins, S. Cleary y J. J. Saleem, «Wearable Heart Rate Monitor Technology Accuracy in,» de *Proceedings of the Human Factors and Ergonomics Society 2017 Annual Meeting*, 2017.
- [32] S. LeBoeuf, «Five Challenges of Optical Heart Rate Monitoring,» Febrero 2016. [En línea]. Available: <https://www.sensorsmag.com/components/five-challenges-optical-heart-rate-monitoring>.
- [33] D. Teichman, C. Brüser, B. Eilebrecht, A. Abbas y N. Blau, «Non-contact Monitoring Techniques - Principles and Applications,» de *34th Annual International Conference of the IEEE EMBS*, San Diego, California USA, 2012.
- [34] J. Gomez-Clapers, A. Serra-Rocamora, R. Casanella y R. Pallas-Areny, «Towards the standardization of ballistocardiography systems for J-peak timing measurement,» *Measurement*, vol. 58, pp. 310-316, 2014.
- [35] L. Dziuda, F. W. Skibniewski, M. Krej y P. M. Baran, «Fiber Bragg grating-based sensor for monitoring respiration and heart activity during magnetic resonance imaging examinations,» *Journal of Biomedical Optics*, vol. 18, nº 5, 2013.
- [36] J. C. Dainty, A. E. Ennos, M. G. J. W. Françon, T. S. McKenzie y G. Parry, *Laser Speckle and Related Phenomena*, Berlin: Springer Berlin Heidelberg, 1975.
- [37] A. Labeyrie, «High-resolution techniques in optical astronomy,» *Prog. Opt.* , vol. 14, nº 47, 1976.
- [38] M. A. Davis, S. M. S. Kazmi y A. K. Dunn, «Imaging depth and multiple scattering in laser speckle contrast imaging,» *Journal of Biomedical Optics*, vol. 19, nº 8, 2014.
- [39] J. Senarathna, A. Rege, N. Li y N. V. Thakor, «Laser Speckle Contrast Imaging: Theory, Instrumentation and Applications,» *IEEE Reviews in Biomedical Engineering*, vol. 6, 2013.
- [40] P. J. Pinzón, D. S. Montero, A. Tapetado y C. Vázquez, «Dual-Wavelength Speckle-Based SI-POF Sensor for Cost-Effective Detection of Microvibrations,» *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 3, nº 2, 2017.
- [41] A. Rodríguez-Cuevas, E. Real Peña, L. Rodríguez-Cobo, M. Lomer y J. M. L. Higuera, «Low-cost fiber specklegram sensor for noncontact continuous patient monitoring,» *Journal of Biomedical Optics*, vol. 22, nº 3, 2017.

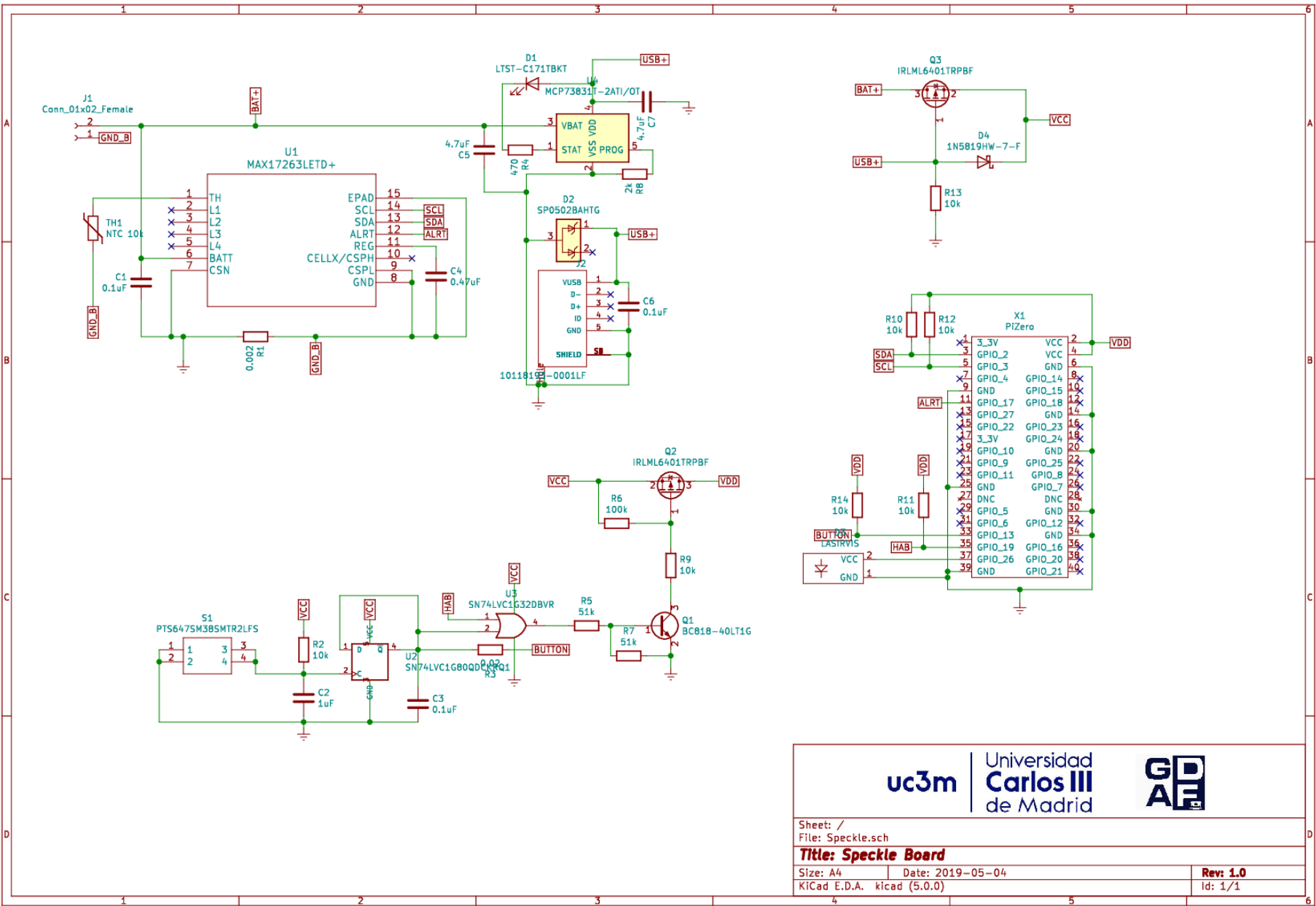
- [42] L. Rodríguez-Cobo, A. Rodríguez-Cuevas, E. Real Peña, J. M. L. Higuera y M. Lomer, «POF-based specklegram sensor for continuous patient monitoring,» de *The 25th International Conference on Plastic Optical Fibers*, Birmingham, UK, 2016.
- [43] P. Podbreznik, D. Donlagic, D. Lesnik y B. Cigale, «Cost-efficient speckle interferometry with plastic optical fiber for unobtrusive monitoring of human vital sign,» *Journal of Biomedical Optics*, vol. 18, nº 10, 2013.
- [44] I. Sanchez, A. Robledo, J. D. López, P. J. Pinzón y C. Vázquez, «POF heart rate detector distribution study,» de *XI Reunión Española de Optoelectrónica*, Zaragoza, 2019.
- [45] L. Thevenaz, *Advanced Fiber Optics*, Laussane: EPFL Press, 2011.
- [46] A. Venkatanarayanan y E. Spain, «Review of Recent Developments of Sensing Materials,» *Reference Module in Materials Science and Materials Engineering*, vol. 13, pp. 47-101, 2014.
- [47] P. Polishuk, «Plastic optical fibers branch out,» *IEEE Communications Magazine*, vol. 44, nº 9, pp. 140-138, 2006.
- [48] FiberLabs Inc., «Stimulated and spontaneous emission,» [En línea]. Available: <https://www.fiberlabs.com/glossary/stimulated-emission/>.
- [49] Thorlabs, «PM20A - Fiber Optic Power Meter, 400 nm - 1100 nm, -60 dBm to +16 dBm (1 nW - 40 mW),» [En línea]. Available: <https://www.thorlabs.com/thorproduct.cfm?partnumber=PM20A#ad-image-0>.
- [50] Amazon, «Akozon módulo láser, 5mW, 650nm,» [En línea]. Available: https://www.amazon.es/gp/product/B07G98HWBD/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1#detail_bullets_id.
- [51] Correlated Solutions, «Application Note AN-1701: Speckle Pattern Fundamentals,» [En línea]. Available: <https://www.correlatedsolutions.com/support/index.php?/Knowledgebase/Article/GetAttachment/80/14750>.
- [52] Adafruit, «TTL Serial Camera,» [En línea]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/ttl-serial-camera.pdf>.
- [53] Raspberry Pi, «Camera Module v2,» [En línea]. Available: <https://www.raspberrypi.org/products/camera-module-v2/>.
- [54] Digi-Key, «TSL3301CL,» [En línea]. Available: <https://www.digikey.es/product-detail/es/ams/TSL3301CL/TSL3301-CLTR-ND/3095203>.
- [55] Digi-Key, «TSL1401CL,» [En línea]. Available: <https://www.digikey.es/product-detail/es/ams/TSL1401CL/TSL1401CLDKR-ND/3095403>.
- [56] Raspberry Pi, «Schematics - Raspberry Pi Documentation,» [En línea]. Available: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md>.

- [57] Arsenijs, «A guide on powering Pi Zero directly from LiIon batteries,» 2017. [En línea]. Available: <https://hackaday.io/project/19035-zero-phone-a-raspberry-pi-smartphone/log/69596-a-guide-on-powering-pi-zero-directly-from-liion-batteries>.
- [58] Wikipedia, «Samsung Galaxy Note 7 - Battery Faults,» [En línea]. Available: https://en.wikipedia.org/wiki/Samsung_Galaxy_Note_7#Battery_faults.
- [59] Batteries 18650, «18650 Battery size-dimensions,» 2013. [En línea]. Available: <http://www.batteries18650.com/2012/12/18650-li-ion-battery-what-all-these.html>.
- [60] Farnell, «SR674361P - Li-Polyme battery datasheet,» [En línea]. Available: http://www.farnell.com/datasheets/2369105.pdf?_ga=2.254647169.887739207.1555670387-1310292400.1555670387.
- [61] Microchip, «MCP73831/2 Datasheet,» [En línea]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/20001984g.pdf>.
- [62] Maxim Integrated, «5.1µA 1-Cell Fuel Gauge with ModelGauge m5 EZ and Optional High-Side Current Sensing,» [En línea]. Available: <https://www.maximintegrated.com/en/products/power/battery-management/MAX17260.html>.
- [63] Maxim Integrated, «Industry's Smallest, Stand-Alone, Single/Multi-Cell Li+ Battery Fuel Gauge to Integrate LED Drivers and a Pushbutton,» [En línea]. Available: <https://www.maximintegrated.com/en/products/power/battery-management/MAX17263.html>.
- [64] Texas Instruments, «SN74LVC1G80-Q1 Single Positive-Edge-Triggered D-Type Flip-Flop,» [En línea]. Available: <http://www.ti.com/lit/ds/symlink/sn74lvc1g80-q1.pdf>.
- [65] Texas Instruments, «SN74LVC1G32 Single 2-Input Positive-OR Gate,» [En línea]. Available: <http://www.ti.com/lit/ds/symlink/sn74lvc1g32.pdf>.
- [66] Maxim Integrated, «MAX17263 Datasheet,» [En línea]. Available: <https://datasheets.maximintegrated.com/en/ds/MAX17263.pdf>.
- [67] KiCad, «KiCad EDA,» [En línea]. Available: <http://kicad-pcb.org/>.
- [68] FreeCAD, «FreeCAD: Your own 3D parametric modeler,» [En línea]. Available: <https://www.freecadweb.org/>.
- [69] Superrei, «One Piece Raspberry Pi Zero + Camera Case (with GPIO),» [En línea]. Available: <https://www.thingiverse.com/thing:1595429>.
- [70] K. Reitz y T. Schlusser, «Scientific Applications,» de *The Hitchhiker's Guide to Python*, O'Reilly Media, 2016.
- [71] D. Jones, «The PiCamera Class,» [En línea]. Available: https://picamera.readthedocs.io/en/release-1.13/api_camera.html#picamera.PiCamera.awb_gains.
- [72] N. Viet Hoan, J.-H. Park, S.-H. Lee y K.-R. Kwon, «Real-time Heart Rate Measurement based on Photoplethysmography using Android Smartphone Camera,» *Journal of Korea Multimedia Society*, pp. 234-243, 2017.

-
- [73] SciPy.org, «[scipy.signal.correlate2d](https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.signal.correlate2d.html) - SciPy v0.10.0 Reference Guide,» [En línea]. Available: <https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.signal.correlate2d.html> .
- [74] Wikipedia, «Filtro de Butterworth,» [En línea]. Available: https://es.wikipedia.org/wiki/Filtro_de_Butterworth.
- [75] Wikipedia, «Finite Impulse Response,» [En línea]. Available: https://en.wikipedia.org/wiki/Finite_impulse_response .
- [76] R. W. Schafer, «What Is a Savitzky-Golay Filter?,» *IEEE Signal Processing Magazine*, pp. 111-117, 2011.
- [77] J. O. Smith, «Spectograms,» de *Mathematics of the Discrete Fourier Transform (DFT), with Audio Applications*, Stanford University, 2007.
- [78] P. D. Welch, «The use of FFT for the estimation of power spectra: a method based on time averaging over short, modified periodograms.,» *IEEE Transactions on Audio and Electroacoustics*, vol. 15, n° 2, pp. 70-73, 1967.
- [79] SciPy.org, «[scipy.signal.find_peaks](https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html) - SciPy v1.2.1 Reference Guide,» [En línea]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html.
- [80] Python, «multiprocessing - Process-based "threading" interface,» [En línea]. Available: <https://docs.python.org/2/library/multiprocessing.html>.
- [81] Python, «multiprocessing - Queue,» [En línea]. Available: <https://docs.python.org/2/library/multiprocessing.html#multiprocessing.Queue>.
- [82] BlueZ, «BlueZ - Official Linux Bluetooth protocol stack,» [En línea]. Available: <http://www.bluez.org/>.
- [83] kernel.org, «[test - bluez.git - Bluetooth protocol stack for Linux,»](https://git.kernel.org/pub/scm/bluetooth/bluez.git/tree/test) [En línea]. Available: <https://git.kernel.org/pub/scm/bluetooth/bluez.git/tree/test>.
- [84] freedesktop.org, «dbus,» [En línea]. Available: <https://www.freedesktop.org/wiki/Software/dbus/>.
- [85] Bluetooth, «GATT Services - Bluetooth Technology Website,» [En línea]. Available: <https://www.bluetooth.com/specifications/gatt/services/>.
- [86] Nordic Semiconductor, «nRF Connect for Mobile,» [En línea]. Available: https://www.nordicsemi.com/?sc_itemid=%7B41FF7A0B-B565-420A-95B7-B32122B5D3AD%7D.
- [87] Nordic Semiconductor, «Android-Scanner-Compat-Library: A compat library for Bluetooth Low Energy scanning on Android.,» [En línea]. Available: <https://github.com/NordicSemiconductor/Android-Scanner-Compat-Library>.
- [88] jjoe64, «[GraphView: Android Graph Library for creating zoomable and scrollable line and bar graphs.,»](https://github.com/jjoe64/GraphView) [En línea]. Available: <https://github.com/jjoe64/GraphView>.
- [89] J. Wharton, «Butter Knife,» [En línea]. Available: <https://jakewharton.github.io/butterknife/>.

-
- [90] Maxim Integrated, «MAX1726x Software Implementation Guide,» [En línea]. Available: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/6595>.
 - [91] R. Krohn, D. Metcalf y P. Salber, Connected Health, CRC Press, 2017.
 - [92] American Heart Association, «Implantable Cardioverter Defibrillator (ICD),» [En línea]. Available: <https://www.heart.org/en/health-topics/arrhythmia/prevention--treatment-of-arrhythmia/implantable-cardioverter-defibrillator-icd>.
 - [93] O. Peltola, Introduction to Wearable Healthcare Technology, 2017.
 - [94] Statista, «Digital Health,» 2018.
 - [95] Statista, «eHealth Report Spain 2018,» 2018.

Anexo I. Esquemático PCB



Anexo II. Layout PCB

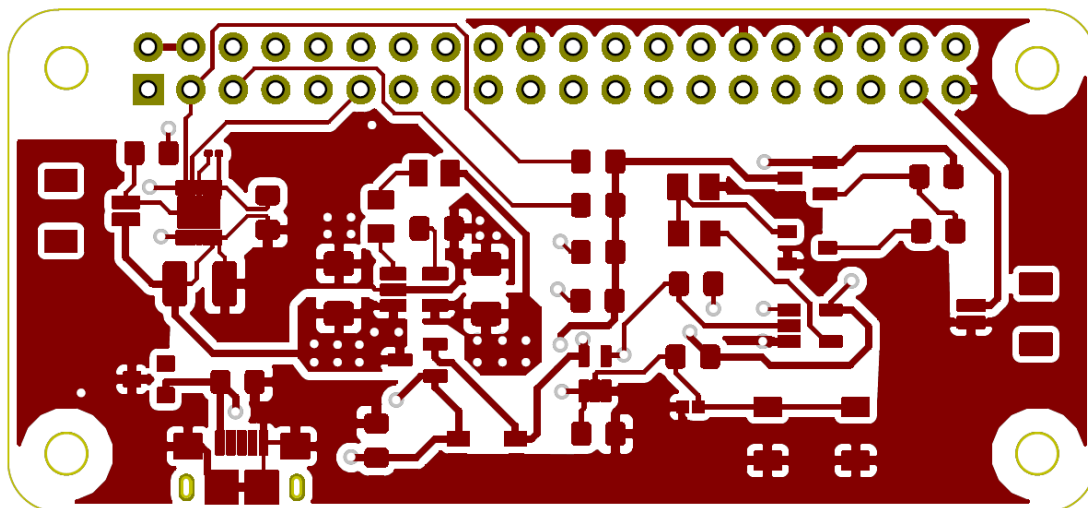


Figura 71. Capa de cobre delantera.

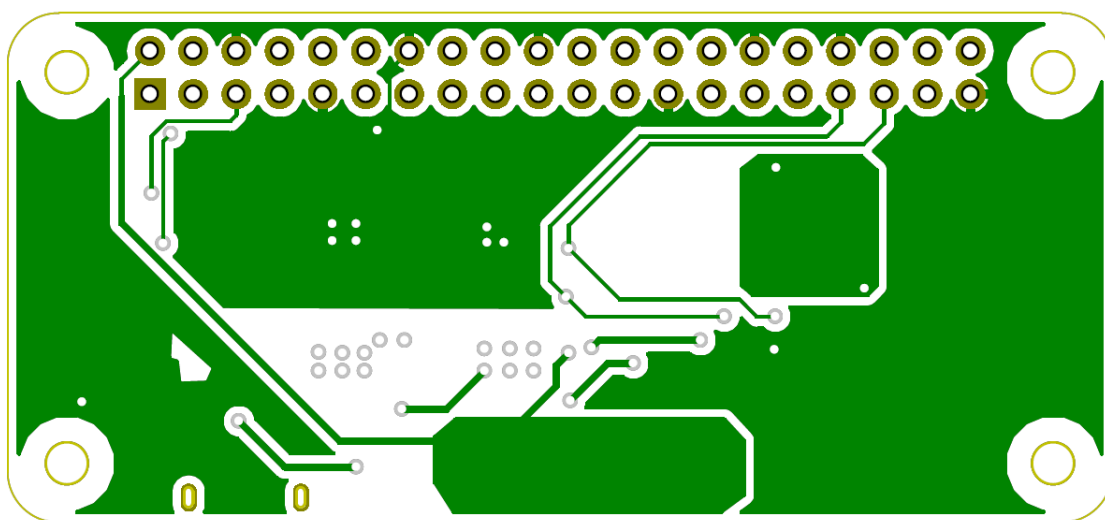


Figura 72. Capa de cobre trasera.

Anexo III. Comparativa procesadores

		Fabricante	Consumo máximo (mA)	Consumo máximo (uA/MHz)	RAM	Frecuencia de reloj (MHz)	Dryhstone MIPS (Benchmark)	Precio	Conectividad	Comunidad	Software desarrollo (familiaridad)	Tamaño (si existe placa desarrollo o evaluación) (mm)	Soporte para cámara	Tiempo de desarrollo	
														IC	Placa evaluación
SBC	Raspberry Pi Zero W	Raspberry Pi Foundation	300	300	512 MB	1000	871,43	21.24 €	WiFi Bluetooth 4.0 USB OTG SPI, I2C, UART		Linux (Raspbian)	65 x 30		Desarrollo Software Placa de expansión (PCB con cámara, láser, batería) Puesta a punto	
	Banana Pi Zero	Banana Pi	700	700	512 MB	1000	-	26 €	WiFi Bluetooth 4.0 USB OTG SPI, I2C, UART		Linux & Android	60 x 30		Desarrollo Software Placa de expansión (PCB con cámara, láser, batería) Puesta a punto	
	PocketBeagle	Circuitco LLC	460	460	512 MB	1000	1704,46*	23,78 €	USB OTG SPI, I2C, UART		Linux (Ubuntu)	56 x 35		Desarrollo Software Desarrollo controlador cámara Placa de expansión (+ Bluetooth) Puesta a punto	
	Carambola 2	8devices	75	187,5	64 MB	400	-	\$25 US	USB Ethernet SPI, I2C, UART		Linux (OpenWRT)	38 x 28		Familiarización Desarrollo Software Placa de expansión (+ Bluetooth) Puesta a punto	
	LinkIt Smart 7688 Duo	Seeed Studio	870	1500	128 MB	580	-	18,28 €	WiFi USB Ethernet JTAG, SPI, I2C, UART		Linux (OpenWRT)	55,7 x 26		Familiarización Desarrollo Software Placa de expansión Puesta a punto	
	Onion Omega 2	Onion	580	1000	128 MB	580	-	14,12 €	WiFi USB Ethernet SPI, I2C, UART		Linux (LEDE**)	42,9 x 26,4		Familiarización Desarrollo Software Placa de expansión (+ Bluetooth) Puesta a punto	

	NanoPi NEO Air	FriendlyARM	300	250	512 MB	1200	-	\$28 US	WiFi Bluetooth USB OTG SPI, I2C, UART		Linux (Debian)	40 x 40		Desarrollo Software Placa de expansión Puesta a punto	
MCUs/MPUs de bajo consumo	SAMA5D3	Microchip	152.2	284	128 KB Interfases DDR2	536	599	10 €	USB Ethernet SPI, I2C, UART		Free Linux	NA		Diseño PCB completo Familiarización Desarrollo software Puesta a punto	
	STM32L5	ST Microelectronics	10.56	96	256 KB	110	165	NA	USB SPI, I2C, UART		Free Linux	NA		Diseño PCB completo (incluir interfaz para la cámara) Familiarización Desarrollo software Puesta a punto	
	STM32L4	ST Microelectronics	7.28	91	256 KB + 64 KB	80	100	IC 10 € PCB 60 €	USB OTG CAN SPI, I2C, UART		C/C++	NA		Diseño PCB completo Familiarización Desarrollo software Puesta a punto	Familiarización Desarrollo Software Placa de expansión Puesta a punto
	PSoC	Cypress	6	40	288 KB	150	187,5	IC 12,44 € PCB 17,34 €	Bluetooth LE 5.0 USB SPI, I2C,UART		PSoC Creator (C)	NA (aprox 30 x 100)		Diseño PCB completo (incluir interfaz para la cámara) Familiarización Desarrollo software Puesta a punto	Familiarización Desarrollo Software Desarrollo controlador cámara Placa de expansión Puesta a punto
	MCF5301x	NXP	82.8	345	128 KB Interfases DDR	240	504	11,82 €	USB OTG Ethernet SPI, I2C, UART		µClinux	-		Diseño PCB completo (incluir interfaz para la cámara) Familiarización Desarrollo software Puesta a punto	

	MSP430	Texas Instruments	1.52	290	32 KB	8	660	IC 5,33 € PCB 11,72€	I2C,SPI,UART		C	NA	Diseño PCB completo (incluir interfaz para la cámara) Desarrollo software Puesta a punto	Familiarización Desarrollo Software Desarrollo controlador cámara Placa de expansión Puesta a punto
DSPs	dsPIC33E	Microchip	60	1000	52 KB	60	-	IC 6,64 € PCB 56,48 €	USB CAN I2C,SPI,UART		MPLAB	NA	Diseño PCB completo (incluir interfaz para la cámara) Desarrollo software Puesta a punto	Desarrollo Software Desarrollo controlador cámara Placa de expansión Puesta a punto
	TMS320	Texas Instruments	4.5	45	64 KB DARAM (Dual) 256 KB SARAM (Single) Externa hasta 16 MB	150	151***	8,41 €	USB I2C,SPI,UART		C	-	Diseño PCB completo (incluir interfaz para la cámara) Desarrollo software Puesta a punto	

*

Datos de BeagleBone Black (mismo procesador)

**

Linux Embedded Development Environment

Calculado a mano a partir de tiempo mínimo de instrucción (6.6 ns)

Suponiendo Vs=3.3V, 0.15mW/MHz